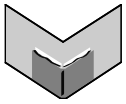
The Journal of Machine Learning Research
Volume 16
Print-Archive Edition

Pages 1–1304



Microtome Publishing
Brookline, Massachusetts
www.mtome.com

The Journal of Machine Learning Research
Volume 16
Print-Archive Edition

The Journal of Machine Learning Research (JMLR) is an open access journal. All articles published in JMLR are freely available via electronic distribution. This Print-Archive Edition is published annually as a means of archiving the contents of the journal in perpetuity. The contents of this volume are articles published electronically in JMLR in 2015.

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

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

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

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

JMLR Editorial Board

Journal of Machine Learning Research

Volume 16, 2016

Journal of Machine Learning Research 16 (2015) 1-46 Submitted 1/14; Revised 7/14; Published 1/15

Statistical Decision Making for Optimal Budget Allocation in Crowd Labeling

Xi Chen XICHEN@NYU.EDU
Stern School of Business
New York University
New York, New York, 10012, USA

Qihang Lin QIHANG-LIN@UIOWA.EDU
Tippie College of Business
University of Iowa
Iowa City, Iowa, 52242, USA

Dengyong Zhou DENGYONG.ZHOU@MICROSOFT.COM
Microsoft Research
Redmond, Washington, 98052, USA

Editor: Yuan (Alan) Qi

Abstract

It has become increasingly popular to obtain machine learning labels through commercial crowdsourcing services. The crowdsourcing workers or annotators are paid for each label they provide, but the task requester usually has only a limited amount of the budget. Since the data instances have different levels of labeling difficulty and the workers have different reliability for the labeling task, it is desirable to wisely allocate the budget among all the instances and workers such that the overall labeling quality is maximized. In this paper, we formulate the budget allocation problem as a Bayesian Markov decision process (MDP), which simultaneously conducts learning and decision making. The optimal allocation policy can be obtained by using the dynamic programming (DP) recurrence. However, DP quickly becomes computationally intractable when the size of the problem increases. To solve this challenge, we propose a computationally efficient approximate policy which is called optimistic knowledge gradient. Our method applies to both pull crowdsourcing marketplaces with homogeneous workers and push marketplaces with heterogeneous workers. It can also incorporate the contextual information of instances when they are available. The experiments on both simulated and real data show that our policy achieves a higher labeling quality than other existing policies at the same budget level.

Keywords: crowdsourcing, budget allocation, Markov decision process, dynamic programming, optimistic knowledge gradient

1. Introduction

In many real applications, data are usually collected without any innate label. For example, a digital camera will not automatically tag a picture as a portrait or a landscape. A traditional approach for data labeling is to hire a small group of experts to provide labels for the entire set of data. However, for large-scale data, such an approach becomes inefficient and very costly. Thanks to the advent of many online crowdsourcing services, e.g., Amazon

Mechanical Turk, a much more efficient way is to post unlabeled data to a crowdsourcing marketplace, where a big crowd of low-paid workers can be hired instantaneously to perform labeling tasks.

Despite its high efficiency and immediate availability, crowd labeling raises many new challenges. Since labeling tasks are tedious and workers are usually non-experts, labels generated by the crowd suffer from low quality. As a remedy, most crowdsourcing services resort to labeling redundancy to reduce the labeling noise, which is achieved by collecting multiple labels from different workers for each data instance. In particular, a crowd labeling process can be described as a two phase procedure:

1. In the first phase, unlabeled data instances are assigned to a crowd of workers and multiple raw labels are collected for each data instance.

2. In the second phase, for each data instance, one aggregates the collected raw labels to infer its true label.

In principle, more raw labels will lead to a higher chance of recovering the true label. However, each raw label comes with a cost: the requester has to pay workers pre-specified monetary reward for each label they provide, usually, regardless of the label's correctness. For example, a worker typically earns 10 cents by categorizing a website as porn or not. In practice, the requester has only a limited amount of budget which essentially restricts the total number of raw labels that he/she can collect. This raises a challenging question central in crowd labeling: *What is the best way to allocate the budget among data instances and workers so that the overall accuracy of aggregated labels is maximized ?*

The most important factors that decide how to allocate the budget are the intrinsic characteristics of data instances and workers: *labeling difficulty/ambiguity for each data instance* and *reliability/quality of each worker*. In particular, an instance is less ambiguous if its label can be decided based on the common knowledge and a vast majority of reliable workers will provide the same label for it. In principle, we should avoid spending too much budget on those easy instances since excessive raw labels will not bring much additional information. In contrast, for an ambiguous instance which falls near the boundary of categories, even those reliable workers will still disagree with each other and generate inconsistent labels. For those ambiguous instances, we are facing a challenging decision problem on how much budget that we should spend on them. On one hand, it is worth to collect more labels to boost the accuracy of the aggregate label. On the other hand, since our goal is to maximize the *overall* labeling accuracy, when the budget is limited, we should simply put those few highly ambiguous instances aside to save budget for labeling less difficult instances. In addition to the ambiguity of data instances, the other important factor is the reliability of each worker and, undoubtedly, it is desirable to assign more instances to those reliable workers. Despite their importance in deciding how to allocate the budget, both the data ambiguity and workers' reliability are unknown parameters at the beginning and need to be updated based on the stream of collected raw labels in an online fashion. This further suggests that the budget allocation policy should be dynamic and simultaneously conduct parameter estimation and decision making.

To search for an optimal budget allocation policy, we model the data ambiguity and workers' reliability using two sets of random variables drawn from known prior distributions. Then, we formulate the problem into a finite-horizon Bayesian Markov Decision

Process (MDP) (Puterman, 2005), whose state variables are the posterior distributions of these variables, which are updated by each new label. Here, the Bayesian setting is necessary. We will show that an optimal policy only exists in the Bayesian setting. Using the MDP formulation, the optimal budget allocation policy for any finite budget level can be readily obtained via the dynamic programming (DP). However, DP is computationally intractable for large-scale problems since the size of the state space grows exponentially in budget level. The existing widely-used approximate policies, such as approximate Gittins index rule (Gittins, 1989) or knowledge gradient (KG) (Gupta and Miescke, 1996; Frazier et al., 2008), either has a high computational cost or poor performance in our problem. In this paper, we propose a new policy, called *optimistic knowledge gradient (Opt-KG)*. In particular, the Opt-KG policy dynamically chooses the next instance-worker pair based on the optimistic outcome of the marginal improvement on the accuracy, which is a function of state variables. We further propose a more general Opt-KG policy using the conditional value-at-risk measure (Rockafellar and Uryasev, 2002). The Opt-KG is computationally efficient, achieves superior empirical performance and has some asymptotic theoretical guarantees.

To better present the main idea of our MDP formulation and the Opt-KG policy, we start from the binary labeling task (i.e., providing the category, either positive or negative, for each instance). We first consider the *pull marketplace* (e.g., Amazon Mechanical Turk or Galaxy Zoo) , where the labeling requester can only post instances to the general worker pool with either anonymous or transient workers, but cannot assign to an identified worker. In a pull marketplace, workers are typically treated as *homogeneous* and one models the entire worker pool instead of each individual worker. We further assume that workers are fully reliable (or noiseless) such that the chance that they make an error only depend on instances' own ambiguity. At a first glance, such an assumption may seem oversimplified. In fact, it turns out that the budget-optimal crowd labeling under such an assumption has been highly non-trivial. We formulate this problem into a Bayesian MDP and propose the computational efficient Opt-KG policy. We further prove that the Opt-KG policy in such a setting is asymptotically consistent, that is, when the budget goes to infinity, the accuracy converges to 100% almost surely.

Then, we extend the MDP formulation to deal with *push marketplaces* with *heterogeneous* workers. In a push marketplace (e.g., data annotation team in Microsoft Bing group), once an instance is allocated to an identified worker, the worker is required to finish the instance in a short period of time. Based on the previous model for fully reliable workers, we further introduce another set of parameters to characterize workers' reliability. Then our decision process simultaneously selects the next instance to label and the next worker for labeling the instance according to the optimistic knowledge gradient policy. In fact, the proposed MDP framework is so flexible that we can further extend it to incorporate contextual information of instances whenever they are available (e.g., as in web search and advertising applications discussed in Li et al., 2010) and to handle multi-class labeling.

In summary, the main contribution of the paper consists of the three folds: (1) we formulate the budget allocation in crowd labeling into a MDP and characterize the *optimal* policy using DP; (2) computationally, we propose an efficient approximate policy, optimistic knowledge gradient; (3) the proposed MDP framework can be used as a general framework to address various budget allocation problems in crowdsourcing (e.g., rating and ranking tasks).

The rest of this paper is organized as follows. In Section 2, we first present the modeling of budget allocation process for binary labeling tasks with fully reliable workers and motivate our Bayesian modeling. In Section 3, we present the Bayesian MDP and the optimal policy via DP. In Section 4, we propose a computationally efficient approximate policy, Opt-KG. In Section 5, we extend our MDP to model heterogeneous workers with different reliability. In Section 6, we present other important extensions, including incorporating contextual information and multi-class labeling. In Section 7, we discuss the related works. In Section 8, we present numerical results on both simulated and real data sets, followed by conclusions in Section 9.

2. Binary Labeling with Homogeneous Noiseless Workers

We first consider the budget allocation problem in a pull marketplace with homogeneous noiseless workers for binary labeling tasks. We note that such a simplification is important for investigating this problem, since the incorporation of workers' reliability and extensions to multiple categories become rather straightforward once this problem is correctly modeled (see Section 5 and 6).

Suppose that there are K instances and each one is associated with a latent true label $Z_i \in \{-1, 1\}$ for $1 \leq i \leq K$. Our goal is to infer the set of positive instances, denoted by $H^* = \{i : Z_i = 1\}$. Here, we assume that the homogeneous worker pool is *fully reliable* or *noiseless*. We note that it does not mean that each worker knows the true label Z_i. Instead, it means that fully reliable workers will do their best to make judgments but their labels may be still incorrect due to the instance's ambiguity. Further, we model the *labeling difficulty/ambiguity* of each instance by a latent soft-label θ_i, which can be interpreted as the percentage of workers in the homogeneous noiseless crowd who will label the i-th instance as positive. In other words, if we randomly choose a worker from a large crowd of fully reliable workers, we will receive a positive label for the i-th instance with probability θ_i and a negative label with probability $1 - \theta_i$. In general, we assume the crowd is large enough so that the value of θ_i can be any value in $[0, 1]$. To see how θ_i characterizes the labeling difficulty of the i-th instance, we consider a concrete example where a worker is asked to label a person as adult (positive) or not (negative) based on the photo of that person. If the person is more than 25 years old, most likely, the corresponding θ_i will be close to 1, generating positive labels consistently. On the other hand, if the person is younger than 15, she may be labeled as negative by almost all the reliable workers since θ_i is close to 0. In both of this cases, we regard the instance (person) easy to label since Z_i can be inferred with a high accuracy based on only a few raw labels. On the contrary, for a person is one or two years below or above 18, the θ_i is near 0.5 and the numbers of positive and negative labels become relatively comparable so that the corresponding labeling task is very difficult. Given the definition of soft labels, we further make the following assumption:

Assumption 1 *We assume that the soft-label θ_i is consistent with the true label in the sense that $Z_i = 1$ if and only if $\theta_i \geq 0.5$, i.e., the majority of the crowd are correct, and hence $H^* = \{i : \theta_i \geq 0.5\}$.*

Given the total budget, denoted by T, we suppose that each label costs one unit of budget. As discussed in the introduction, the crowd labeling has two phases. The first

Instance	1st round label	2nd round label
Instance 1 (θ_1)	1	1
Instance 2 (θ_2)	1	-1
Instance 3 (θ_3)	1	

Table 1: Toy example with 3 instances and 5 collected labels. Instance 1 has two positive labels, instance 2 has one positive and one negative label, and instance 3 has only one positive label. The question is that, given only one more labeling chance, which instance should be chosen to label?

phase is the *budget allocation phase*, which is a dynamic decision process with T stages. In each stage $0 \leq t \leq T-1$, an instance $i_t \in \mathcal{A} = \{1, \ldots, K\}$ is selected based on the historical labeling results. Once i_t is selected, it will be labeled by a random worker from the homogeneous noiseless worker pool. According to the definition of θ_{i_t}, the label received, denoted by $y_{i_t} \in \{-1, 1\}$, will follow the Bernoulli distribution with the parameter θ_{i_t}:

$$\Pr(y_{i_t} = 1) = \theta_{i_t} \qquad \text{and} \qquad \Pr(y_{i_t} = -1) = 1 - \theta_{i_t}. \tag{1}$$

We note that, at this moment, all workers are assumed to be homogeneous and noiseless so that y_{i_t} only depends on θ_{i_t} but not on which worker provides the label. Therefore, it is suffice for the decision maker (e.g., requester or crowdsourcing service) to select the instance in each stage instead of an instance-worker pair.

The second phase is the *label aggregation phase*. When the budget is exhausted, the decision maker needs to infer true labels $\{Z_i\}_{i=1}^n$ by aggregating all the collected labels. According to Assumption 1, it is equivalent to infer the set of positive instances whose $\theta_i \geq 0.5$. Let H_T be the estimated positive set. The final overall accuracy is measured by $|H_T \cap H^*| + |(H_T)^c \cap (H^*)^c|$, the size of the mutual overlap between H^* and H_T.

Our goal is to determine the *optimal allocation policy*, $(i_0, \ldots, i_{T-1})$, so that overall accuracy is maximized. Here, a natural question to ask is whether the *optimal* allocation policy exists and what assumptions do we need for the existence of the optimal policy. To answer this question, we provide a concrete example, which motivates our Bayesian modeling.

2.1 Why We Need a Bayesian Modeling

Let us check a toy example with 3 instances and 5 collected labels (see Table 1). We assume that the workers are homogeneous noiseless and the label aggregation is performed by the majority vote rule. *Now if we only have the budget to get one more label, which instance should be chosen to label?* It is obvious that we should not put the remaining budget on the first instance since we are relatively more confident on what its true label should be. Thus, the problem becomes how to choose between the second and third instances. In what follows, we shall show that *there is no optimal policy under the frequentist setting*. To be more explicit, the optimal policy leads to the expected accuracy which is at least as good as that of all other policies for any values of $\{\theta_i\}_{i=1}^n$.

	Current Accuracy	$y=1$	$y=-1$	Expected Accuracy	Improvement
$\theta_1 > 0.5$	1	1	1	1	0
$\theta_1 < 0.5$	0	0	0	0	0
$\theta_2 > 0.5$	0.5	1	0	θ_2	$\theta_2 - 0.5 > 0$
$\theta_2 < 0.5$	0.5	0	1	$1-\theta_2$	$0.5 - \theta_2 > 0$
$\theta_3 > 0.5$	1	1	0.5	$\theta_3 + 0.5(1-\theta_3)$	$0.5(\theta_3 - 1) < 0$
$\theta_3 < 0.5$	0	0	0.5	$0.5(1-\theta_3)$	$0.5(1-\theta_3) > 0$

Table 2: Expected improvements in accuracy for collecting an extra label, i.e., the expected accuracy of obtaining one more label minus the current expected accuracy. The 3rd and 4th columns contain the accuracies with the next label being 1 and -1. The 5th is the expected accuracy which is computed by taking θ times the 3rd column plus $(1-\theta)$ times the 4th. The last column contains the expected improvements which is computed by taking the difference between the 5th and 2nd columns.

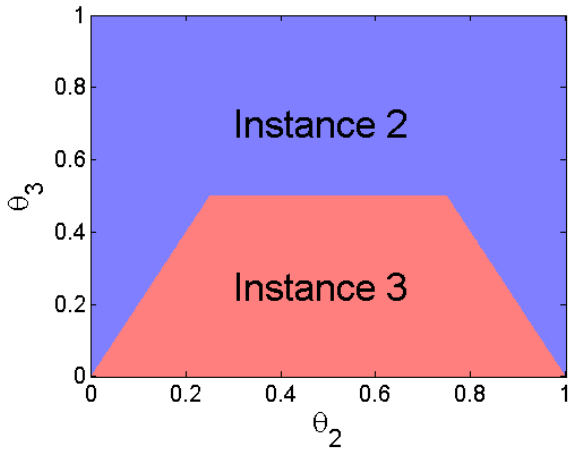


Figure 1: Decision Boundary.

Let us compute the expected improvement in accuracy in terms of the frequentist risk in Table 2. We assume that $\theta_i \neq 0.5$ and if the number of 1 and -1 labels are the same for an instance, the accuracy is 0.5 based on a random guess. From Table 2, we should not label the first instance since the improvement is always 0. This coincides with our intuition. When $\max(\theta_2 - 0.5, 0.5 - \theta_2) > 0.5(1-\theta_3)$ or $\theta_3 > 0.5$, which corresponds to the blue region in Figure 1, we should choose to label the second instance. Otherwise, we should ask the label for the third one. Since the true value of θ_2 and θ_3 are unknown, a optimal policy does not exist under the frequentist paradigm. Further, it will be difficult to estimate θ_2 and θ_3 accurately when the budget is very limited.

In contrast, in a Bayesian setting with prior distribution on each θ_i, the optimal policy is defined as the policy which leads to the highest expected accuracy under the given prior

instead of for any possible values of $\{\theta_i\}_{i=1}^n$. Therefore, we can optimally determine the next instance to label by taking another expectation over the distribution of θ_i. In this paper, we adopt the Bayesian modeling to formulate the budget allocation problem in crowd labeling.

3. Bayesian MDP and Optimal Policy

In this section, we first introduce the Bayesian MDP for modeling the dynamic budget allocation process and then provide the optimal allocation policy using dynamic programming.

3.1 Bayesian Modeling

We assume that each θ_i is drawn from a known Beta prior $\text{Beta}(a_i^0, b_i^0)$. Beta is a rich family of distributions in the sense that it exhibits a fairly wide variety of shapes on the domain of θ_i, i.e., the unit interval $[0, 1]$. For presentation simplicity, instead of considering a full Bayesian model with hyper-priors on a_i^0 and b_i^0, we fix a_i^0 and b_i^0 at the beginning. In practice, if the budget is sufficient, one can first label each instance equally many times to pre-estimate $\{a_i^0, b_i^0\}_{i=1}^K$ before the dynamic labeling procedure is invoked. Otherwise, when there is no prior knowledge, we can simply assume $a_i^0 = b_i^0 = 1$ so that the prior is a uniform distribution. According to our simulated experimental results in Section 8.1.2, uniform prior works reasonably well unless the data is highly skewed in terms of class distribution. Other commonly used uninformative priors such as Jeffreys prior or reference prior ($\text{Beta}(1/2, 1/2)$) or Haldane prior ($\text{Beta}(0, 0)$) can also be adopted (see Robert, 2007 for more on uninformative priors). Choices of prior distributions are discussed in more details in Section 4.2.

At each stage t with $\text{Beta}(a_i^t, b_i^t)$ as the current posterior distribution for θ_i, we make a decision by choosing an instance $i_t \in \mathcal{A} = \{1, \ldots, K\}$ and acquire its label $y_{i_t} \sim \text{Bernoulli}(\theta_{i_t})$. Here $\mathcal{A}$ denotes the *action set*. By the fact that Beta is the conjugate prior of the Bernoulli, the posterior of θ_{i_t} in the stage $t+1$ will be updated as:

$$\text{Beta}(a_{i_t}^{t+1}, b_{i_t}^{t+1}) = \begin{cases} \text{Beta}(a_{i_t}^t + 1, b_{i_t}^t) & \text{if} \quad y_{i_t} = 1; \\ \text{Beta}(a_{i_t}^t, b_{i_t}^t + 1) & \text{if} \quad y_{i_t} = -1. \end{cases}$$

We put $\{a_i^t, b_i^t\}_{i=1}^K$ into a $K \times 2$ matrix S^t, called a *state matrix*, and let $S_i^t = (a_i^t, b_i^t)$ be the i-th row of S^t. The update of the state matrix can be written in a more compact form:

$$S^{t+1} = \begin{cases} S^t + (\mathbf{e}_{i_t}, \mathbf{0}) & \text{if } y_{i_t} = 1; \\ S^t + (\mathbf{0}, \mathbf{e}_{i_t}) & \text{if } y_{i_t} = -1, \end{cases} \tag{2}$$

where $\mathbf{e}_{i_t}$ is a $K \times 1$ vector with 1 at the i_t-th entry and 0 at all other entries. As we can see, $\{S^t\}$ is a Markovian process because S^{t+1} is completely determined by the current state S^t, the action i_t and the obtained label y_{i_t}. It is easy to calculate the *state transition probability* $\Pr(y_{i_t}|S^t, i_t)$, which is the posterior probability that we are in the next state S^{t+1} if we choose i_t to be label in the current state S^t:

$$\Pr(y_{i_t} = 1|S^t, i_t) = \mathbb{E}(\theta_{i_t}|S^t) = \frac{a_{i_t}^t}{a_{i_t}^t + b_{i_t}^t} \quad \text{and} \quad \Pr(y_{i_t} = -1|S^t, i_t) = \frac{b_{i_t}^t}{a_{i_t}^t + b_{i_t}^t}. \tag{3}$$

Given this labeling process, the budget allocation policy is defined as a sequence of decisions: $\pi = (i_0, \ldots, i_{T-1})$. Here, we require decisions depend only upon the previous information. To make this more formal, we define a filtration $\{\mathcal{F}_t\}_{t=0}^{T}$, where $\mathcal{F}_t$ is the information collected until the stage $t-1$. More precisely, $\mathcal{F}_t$ is the the σ-algebra generated by the sample path $(i_0, y_{i_0}, \ldots, i_{t-1}, y_{i_{t-1}})$. We require the action i_t is determined based on the historical labeling results up to the stage $t-1$, i.e., i_t is $\mathcal{F}_t$-measurable.

3.2 Inference About True Labels

As described in Section 2, the budget allocation process has two phases: the dynamic budget allocation phase and the label aggregation phase. Since the goal of the dynamic budget allocation in the first phase is to maximize the accuracy of aggregated labels in the second phase, we first present how to infer the true label via label aggregation in the second phase.

When the decision process terminates at the stage T, we need to determine a positive set H_T to maximize the *conditional* expected accuracy conditioning on $\mathcal{F}_T$, which corresponds to minimizing the posterior risk:

$$H_T = \underset{H \subset \{1,\ldots,K\}}{\arg\max} \mathbb{E}\left(\sum_{i=1}^{K} \big(\mathbf{1}(i \in H) \cdot \mathbf{1}(i \in H^*) + \mathbf{1}(i \notin H) \cdot \mathbf{1}(i \notin H^*) \big) \middle| \mathcal{F}_T \right), \tag{4}$$

where $\mathbf{1}(A)$ is the indicator function, which takes the value 1 if the event A is true and 0 otherwise. The term inside expectation in (4) is the binary labeling accuracy which can also be written as $|H \cap H^*| + |H^c \cap (H^*)^c|$.

We first observe that, for $0 \le t \le T$, the conditional distribution $\theta_i | \mathcal{F}_t$ is exactly the posterior distribution $\mathrm{Beta}(a_i^t, b_i^t)$, which depends on the historical sampling results only through $S_i^t = (a_i^t, b_i^t)$. Hence, we define

$$I(a, b) \doteq \Pr(\theta \ge 0.5 | \theta \sim \mathrm{Beta}(a, b)), \tag{5}$$

$$P_i^t \doteq \Pr(i \in H^* | \mathcal{F}_t) = \Pr(\theta_i \ge 0.5 | \mathcal{F}_t) = \Pr(\theta_i \ge 0.5 | S_i^t) = I(a_i^t, b_i^t). \tag{6}$$

As shown in Xie and Frazier (2013), the optimal positive set H_T can be determined by the Bayes decision rule as follows.

Proposition 2 *$H_T = \{i : \Pr(i \in H^* | \mathcal{F}_T) \ge 0.5\} = \{i : P_i^T \ge 0.5\}$ solves (4).*

The proof of Proposition 2 is given in the appendix for completeness.

With Proposition 2 in place, we plug the optimal positive set H_T into the right hand side of (4) and the conditional expected accuracy given $\mathcal{F}_T$ can be simplified as:

$$\mathbb{E}\left(\sum_{i=1}^{K} \big(\mathbf{1}(i \in H_T) \cdot \mathbf{1}(i \in H^*) + \mathbf{1}(i \notin H_T) \cdot \mathbf{1}(i \notin H^*) \big) \middle| \mathcal{F}_T \right) = \sum_{i=1}^{K} h(P_i^T), \tag{7}$$

where $h(x) \doteq \max(x, 1-x)$. We also note that P_i^T provides not only the estimated label for the i-th instance but also how confident the estimated label is correct. According to the next corollary with the proof in the appendix, we show that the optimal H_T is constructed based on a refined *majority vote* rule which incorporates the prior information.

Corollary 3 *$I(a,b) > 0.5$ if and only if $a > b$ and $I(a,b) = 0.5$ if and only if $a = b$. Therefore, $H_T = \{i : a_i^T \geq b_i^T\}$ solves (4).*

By viewing a_i^0 and b_i^0 as pseudo-counts of 1s and -1s at the initial stage, the parameters a_i^T and b_i^T are the total counts of 1s and -1s. The estimated positive set $H_T = \{i : a_i^T \geq b_i^T\}$ consists of instances with more (or equal) counts of 1s than that of -1s. When $a_i^0 = b_i^0$, H_T is constructed exactly according to the vanilla *majority vote* rule.

To find the optimal allocation policy which maximizes the expected accuracy, we need to solve the following optimization problem:

$$\begin{aligned} V(S^0) &\doteq \sup_{\pi} \mathbb{E}^{\pi} \left[\mathbb{E} \left(\sum_{i=1}^{K} \big(\mathbf{1}(i \in H_T) \cdot \mathbf{1}(i \in H^*) + \mathbf{1}(i \notin H_T) \cdot \mathbf{1}(i \notin H^*) \big) \middle| \mathcal{F}_T \right) \right] \\ &= \sup_{\pi} \mathbb{E}^{\pi} \left(\sum_{i=1}^{K} h(P_i^T) \right), \end{aligned} \tag{8}$$

where $\mathbb{E}^{\pi}$ represents the expectation taken over the sample paths $(i_0, y_{i_0}, \ldots, i_{T-1}, y_{i_{T-1}})$ generated by a policy π. The second equality is due to Proposition 2 and $V(S^0)$ is called value function at the initial state S^0. The optimal policy π^* is any policy π that attains the supremum in (8).

3.3 Markov Decision Process

The optimization problem in (8) is essentially a Bayesian multi-armed bandit (MAB) problem, where each instance corresponds to an arm and the decision is which instance/arm to be sampled next. However, it is different from the classical MAB problem (Auer et al., 2002; Bubeck and Cesa-Bianchi, 2012), which assumes that each sample of an arm yields independent and identically distributed (*i.i.d.*) reward according to some unknown distribution associated with that arm. Given the total budget T, the goal is to determine a sequential allocation policy so that the collected rewards can be maximized. We contrast this problem with our problem: instead of collecting intermediate independent rewards on the fly, our objective in (8) merely involves the final "reward", i.e., overall labeling accuracy, which is only available at the final stage when the budget runs out. Although there is no intermediate reward in our problem, we can still decompose the final expected accuracy into sum of *stage-wise rewards* using the technique from Xie and Frazier (2013), which further leads to our MDP formulation. Since these *stage-wise rewards* are artificially created, they are no longer i.i.d. for each instance. We also note that the problem in Xie and Frazier (2013) is an infinite-horizon one which optimizes the stopping time while our problem is *finite-horizon* since the decision process must be stopped at the stage T.

Proposition 4 *Define the stage-wise expected reward as:*

$$R(S^t, i_t) = \mathbb{E} \left(\sum_{i=1}^{K} h(P_i^{t+1}) - \sum_{i=1}^{K} h(P_i^t) \middle| S^t, i_t \right) = \mathbb{E} \left(h(P_{i_t}^{t+1}) - h(P_{i_t}^t) | S^t, i_t \right), \tag{9}$$

then the value function (8) becomes:

$$V(S^0) = G_0(S^0) + \sup_{\pi} \mathbb{E}^{\pi} \left(\sum_{t=0}^{T-1} R(S^t, i_t) \right), \tag{10}$$

where $G_0(S^0) = \sum_{i=1}^{K} h(P_i^0)$ *and the optimal policy* π^* *is any policy* π *that attains the supremum.*

The proof of Proposition 4 is presented in the appendix. In fact, the stage-wise reward in (9) has a straightforward interpretation. According to (8), the term $\sum_{i=1}^{K} h(P_i^t)$ is the expected accuracy at the t-th stage. The stage-wise reward $R(S^t, i_t)$ takes the form of the difference between the expected accuracy at the $(t+1)$-stage and the t-th stage, i.e., the expected gain in accuracy for collecting another label for the i_t-th instance. The second equality in (9) holds simply because: only the i_t-th instance receives the new label and the corresponding $P_{i_t}^t$ changes while all other P_i^t remain the same. Since the expected reward (9) only depends on $S_{i_t}^t = (a_{i_t}^t, b_{i_t}^t)$, we write

$$R(S^t, i_t) = R\left(S_{i_t}^t\right) = R\left(a_{i_t}^t, b_{i_t}^t\right), \tag{11}$$

and use them interchangeably. The function $R(a, b)$ with two parameters a and b has an analytical representation as follows. For any state (a, b) of a single instance, the reward of getting a label 1 and a label -1 are:

$$R_1(a, b) = h(I(a+1, b)) - h(I(a, b)), \tag{12}$$
$$R_2(a, b) = h(I(a, b+1)) - h(I(a, b)). \tag{13}$$

The expected reward takes the following form:

$$R(a, b) = p_1 R_1 + p_2 R_2, \tag{14}$$

where $p_1 = \frac{a}{a+b}$ and $p_2 = \frac{b}{a+b}$ are the transition probabilities in (3).

With Proposition 4, the maximization problem (8) is formulated as a T-stage *Markov Decision Process* (MDP) as in (10), which is associated with a tuple:

$$\{T, \{\mathcal{S}^t\}, \mathcal{A}, \Pr(y_{i_t} | S^t, i_t), R(S^t, i_t)\}.$$

Here, the state space at the stage t, $\mathcal{S}^t$, is all possible states that can be reached at t. Once we collect a label y_{i_t}, one element in S^t (either $a_{i_t}^t$ or $b_{i_t}^t$) will add one. Therefore, we have

$$\mathcal{S}^t = \left\{ \{a_i^t, b_i^t\}_{i=1}^K : a_i^t \geq a_i^0, b_i^t \geq b_i^0, \sum_{i=1}^{K} (a_i^t - a_i^0) + (b_i^t - b_i^0) = t \right\}. \tag{15}$$

The action space is the set of instances that could be labeled next: $\mathcal{A} = \{1, \ldots, K\}$. The transition probability $\Pr(y_{i_t} | S^t, i_t)$ is defined in (3) and the expected reward at each stage $R(S^t, i_t)$ is defined in (9).

Remark 5 *We can also view Proposition 4 as a consequence of applying the reward shaping technique (Ng et al., 1999) to the original problem* (8)*. In fact, we can add an artificial*

Instance i	S_i^{T-1}	p_1	p_2	$R_1(S_i^{T-1})$	$R_2(S_i^{T-1})$	$R(S^{T-1}, i) = R(S_i^{T-1})$
1	(3,1)	$\frac{3}{4}$	$\frac{1}{4}$	0.0625	-0.1875	$\frac{3}{4} \cdot (0.0625) + \frac{1}{4} \cdot (-0.1875) = 0$
2	(2,2)	$\frac{1}{2}$	$\frac{1}{2}$	0.1875	0.1875	$\frac{1}{2} \cdot (0.1875) + \frac{1}{2} \cdot (0.1875) = 0.1875$
3	(2,1)	$\frac{2}{3}$	$\frac{1}{3}$	0.1250	-0.2500	$\frac{2}{3} \cdot (0.1250) + \frac{1}{3} \cdot (-0.2500) = 0$

Table 3: Calculation of the expected reward for the toy example in Table 1 according to (12), (13) and (14).

absorbing state, named S_{obs}, to the original state space (15) *and assume that, when the budget allocation process finishes, the state must transit one more time to reach S_{obs} regardless of which action is taken. Hence, the original problem* (8) *becomes a MDP that generates a zero transition reward until the state enters S_{obs} where the transition reward is $\sum_{i=1}^{K} h(P_i^T)$. Then, we define a potential-based shaping function (Ng et al., 1999) over this extended state space as $\Phi(S^t) = \sum_{i=1}^{K} h(P_i^t)$ for $S^t \in \mathcal{S}^t$ and $\Phi(S_{obs}) = 0$. After this,* (4) *can be viewed as a new MDP whose transition reward equals that of* (8) *plus the shaping-reward function $\Phi(S') - \Phi(S)$ when the state transits from S to S'. According to Theorem 1 in Ng et al. (1999),* (4) *and* (8) *have the same optimal policy. This provides an alternative justification for Proposition 4.*

3.4 Optimal Policy via DP

With the MDP in place, we can apply the dynamic programming (DP) algorithm (a.k.a. backward induction) (Puterman, 2005) to compute the optimal policy:

1. Set $V_{T-1}(S^{T-1}) = \max_{i \in \{1,\dots,K\}} R(S^{T-1}, i)$ for *all possible states* $S^{T-1} \in \mathcal{S}^{T-1}$. The optimal decision $i^*_{T-1}(S^{T-1})$ is the decision i that achieves the maximum when the state is S^{T-1}.

2. Iterate for $t = T-2, \dots, 0$, compute the $V_t(S^t)$ for all possible $S^t \in \mathcal{S}^t$ using the Bellman equation:

$$\begin{aligned} &V_t(S^t) \\ &= \max_i \Big(R(S^t, i) + \Pr(y_i = 1 | S^t, i) V_{t+1}\left(S^t + (\mathbf{e}_i, \mathbf{0})\right) + \Pr(y_i = -1 | S^t, i) V_{t+1}\left(S^t + (\mathbf{0}, \mathbf{e}_i)\right) \Big), \end{aligned}$$

   and $i_t^*(S^t)$ is the i that achieves the maximum.

The optimal policy $\pi^* = (i_0^*, \dots, i_T^*)$. For an illustration purpose, we use DP to calculate the optimal instance to be labeled next in the toy example in Section 2.1 under the uniform prior $B(1,1)$ for all θ_i. Since we assume that there is only one labeling chance remaining, which corresponds to the last stage of DP, we should choose the instance $i^*_{T-1}(S^{T-1}) = \arg\max_{i \in \{1,\dots,K\}} R(S^{T-1}, i)$. According to the calculation in Table 3, there is a unique optimal instance for labeling, which is the second instance.

Although DP finds the optimal policy, its computation is intractable since the size of the state space $|\mathcal{S}^t|$ grows exponentially in t according to (15). Therefore, we need to develop a computationally efficient approximate policy, which is the goal of the next section.

4. Approximate Policies

Since DP is computationally intractable, approximate policies are needed for large-scale applications. The simplest policy is the uniform sampling (a.k.a, pure exploration), i.e., we choose the next instance uniformly and independently at random: $i_t \sim \text{Uniform}(1, \ldots, K)$. However, this policy does not explore any structure of the problem.

With the decomposed reward function, our problem is essentially a finite-horizon Bayesian MAB problem. Gittins (1989) showed that Gittins index policy is optimal for infinite-horizon MAB with the discounted reward. It has been applied to the infinite-horizon version of problem (10) in Xie and Frazier (2013). Since our problem is finite-horizon, Gittins index is no longer optimal while it can still provide us a good heuristic index rule. However, the computational cost of Gittins index is very high: the state-of-art-method proposed by Nino-Mora (2011) requires $O(T^6)$ time and space complexity.

A computationally more attractive policy is the knowledge gradient (KG) (Gupta and Miescke, 1996; Frazier et al., 2008). It is essentially a single-step look-ahead policy, which greedily selects the next instance with the largest expected reward:

$$i_t = \underset{i\in\{1,\ldots,K\}}{\arg\max} \left(R(a_i^t, b_i^t) \doteq \frac{a_i^t}{a_i^t + b_i^t} R_1(a_i^t, b_i^t) + \frac{b_i^t}{a_i^t + b_i^t} R_2(a_i^t, b_i^t) \right). \tag{16}$$

As we can see, this policy corresponds to the last stage in DP and hence KG policy is optimal if only one labeling chance is remaining.

When there is a tie, if we select the smallest index i, the policy is referred to *deterministic KG* while if we randomly break the tie, the policy is referred to *randomized KG*. Although KG has been successfully applied to many MDP problems (Powell, 2007), it will fail in our problem as shown in the next proposition with the proof in the appendix.

Proposition 6 *Assuming that a_i^0 and b_i^0 are positive integers and letting $\mathcal{E} = \{i : a_i^0 = b_i^0\}$, then the deterministic KG policy will acquire one label for each instance in $\mathcal{E}$ and then consistently obtain the label for the first instance even if the budget T goes to infinity.*

According to Proposition 6, the deterministic KG is *not* a *consistent policy*, where the consistent policy refers to the policy that will provide correct labels for all instances (i.e., $H_T = H^*$) almost surely when T goes to infinity. We note that randomized KG policy can address this problem. However, from the proof of Proposition 6, randomized KG behaves similarly to the uniform sampling policy in many cases and its empirical performance is undesirable according to Section 8. In the next subsection, we will propose a new approximate allocation policy based on KG which is a consistent policy with superior empirical performance.

4.1 Optimistic Knowledge Gradient

The stage-wise reward can be viewed as a random variable with a two point distribution, i.e., with the probability $p_1 = \frac{a}{a+b}$ of being $R_1(a, b)$ and the probability $p_2 = \frac{b}{a+b}$ of being $R_2(a, b)$. The KG policy selects the instance with the largest *expected* reward. However, it is not consistent.

Algorithm 1 Optimistic Knowledge Gradient

Input: Parameters of prior distributions for instances $\{a_i^0, b_i^0\}_{i=1}^K$ and the budget T.

for $t = 0, \ldots, T-1$ **do**

Select the next instance i_t to label according to:

$$i_t = \underset{i \in \{1,\ldots,K\}}{\arg\max} \left(R^+(a_i^t, b_i^t) \doteq \max(R_1(a_i^t, b_i^t), R_2(a_i^t, b_i^t)) \right). \tag{17}$$

Acquire the label $y_{i_t} \in \{-1, 1\}$.

if $y_{i_t} = 1$ **then**

$a_{i_t}^{t+1} = a_{i_t}^t + 1, b_{i_t}^{t+1} = b_{i_t}^t$; $a_i^{t+1} = a_i^t, b_i^{t+1} = b_i^t$ for all $i \neq i_t$.

else

$a_{i_t}^{t+1} = a_{i_t}^t, b_{i_t}^{t+1} = b_{i_t}^t + 1$; $a_i^{t+1} = a_i^t, b_i^{t+1} = b_i^t$ for all $i \neq i_t$.

end if

end for

Output: The positive set $H_T = \{i : a_i^T \geq b_i^T\}$.

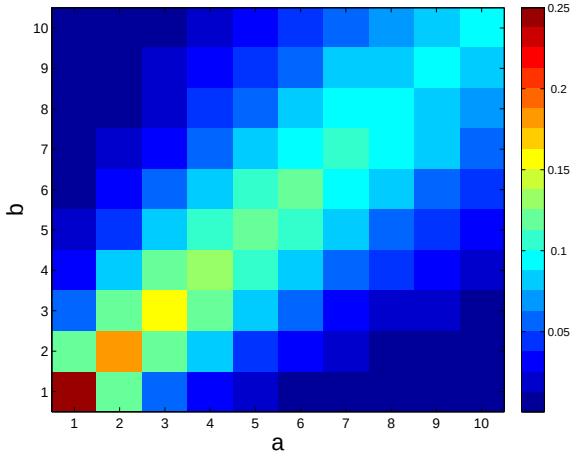


Figure 2: Illustration of $R^+(a, b)$.

In this section, we introduce a new index policy called "optimistic knowledge gradient" (Opt-KG) policy. The Opt-KG policy assumes that decision makers are optimistic in the sense that they select the next instance based on the optimistic outcome of the reward. As a simplest version of the Opt-KG policy, for any state (a_i^t, b_i^t), the optimistic outcome of the reward $R^+(a_i^t, b_i^t)$ is defined as maximum over the reward of obtaining the label 1, $R_1(a_i^t, b_i^t)$, and the reward of obtaining the label -1, $R_2(a_i^t, b_i^t)$. Then the optimistic decision maker selects the next instance i with the largest $R^+(a_i^t, b_i^t)$ as in (17) in Algorithm 1. The overall decision process using the Opt-KG policy is highlighted in Algorithm 1.

In the next theorem, we prove that Opt-KG policy is consistent.

Theorem 7 *Assuming that a_i^0 and b_i^0 are positive integers, the Opt-KG is a consistent policy, i.e, as T goes to infinity, the accuracy will be* 100% *(i.e., $H_T = H^*$) almost surely.*

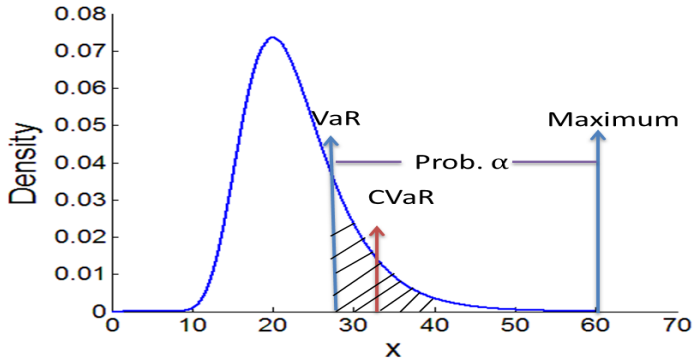


Figure 3: Illustration of Conditional Value-at-Risk.

The key of proving the consistency is to show that when T goes to infinity, each instance will be labeled infinitely many times. We prove this by showing that for any pair of positive integers (a, b), $R^+(a, b) = \max(R_1(a, b), R_2(a, b)) > 0$ and $R^+(a, b) \to 0$ when $a + b \to \infty$. As an illustration, the values of $R^+(a, b)$ are plotted in Figure 2. Then, by strong law of large number, we obtain the consistency of the Opt-KG as stated in Theorem 7. The details are presented in the appendix. We have to note that asymptotic consistency is the minimum guarantee for a good policy. However, it does not necessarily guarantee the good empirical performance for the finite budget level. We will use experimental results to show the superior performance of the proposed policy.

The proposed Opt-KG policy is a general framework for budget allocation in crowd labeling. We can extend the allocation policy based on the maximum over the two possible rewards (Algorithm 1) to a more general policy using the conditional value-at-risk (CVaR) (Rockafellar and Uryasev, 2002). We note that here, instead of adopting the CVaR as a risk measure, we apply it to the reward distribution. In particular, for a random variable X with the support $\mathcal{X}$ (e.g., the random reward with the two point distribution), let α-quantile function be denoted as $Q_\alpha(X) = \inf\{x \in \mathcal{X} : \alpha \leq F_X(x)\}$, where $F_X(\cdot)$ is the CDF of X. The value-at-risk $\text{VaR}_\alpha(X)$ is the smallest value such that the probability that X is less than (or equal to) it is greater than (or equal to) $1 - \alpha$: $\text{VaR}_\alpha(X) = Q_{1-\alpha}(X)$. The conditional value-at-risk ($\text{CVaR}_\alpha(X)$) is defined as the expected reward exceeding (or equal to) $\text{VaR}_\alpha(X)$. An illustration of CVaR is shown in Figure 3.

For our problem, according to Rockafellar and Uryasev (2002), $\text{CVaR}_\alpha(X)$ can be expressed as a simple linear program:

$$\begin{aligned} \text{CVaR}_\alpha(X) = \max_{\{q_1 \geq 0, q_2 \geq 0\}} & \; q_1 R_1 + q_2 R_2, \\ \text{s.t.} \quad & q_1 \leq \frac{1}{\alpha} p_1, \; q_2 \leq \frac{1}{\alpha} p_2, \; q_1 + q_2 = 1. \end{aligned}$$

As we can see, when $\alpha = 1$, $\text{CVaR}_\alpha(X) = p_1 R_1 + p_2 R_2$, which is the expected reward; when $\alpha \to 0$, $\text{CVaR}_\alpha(X) = \max(R_1, R_2)$, which is used as the selection criterion in (17) in Algorithm 1. In fact, a more general Opt-KG policy could be selecting the next instance with the largest $\text{CVaR}_\alpha(X)$ with a tuning parameter $\alpha \in [0, 1]$. We can extend Theorem 7 to prove that the policy based on $\text{CVaR}_\alpha(X)$ is consistent for any $\alpha < 1$. According to our

own experience, $\alpha \to 0$ usually has a better performance in our problem especially when the budget is very limited. Therefore, for the sake of presentation simplicity, we introduce the Opt-KG using $\max(R_1, R_2)$ (i.e., $\alpha \to 0$ in $\text{CVaR}_\alpha(X)$) as the selection criterion.

Finally, we highlight that the Opt-KG policy is computationally very efficient. For K instances with T units of the budget, the overall time and space complexity are $O(KT)$ and $O(K)$ respectively. It is much more efficient that the Gittins index policy which requires $O(T^6)$ time and space complexity.

4.2 Discussions

It is interesting to see the connection between the idea of making the decision based on the optimistic outcome of the reward and the UCB (upper confidence bounds) policy (Auer et al., 2002) for the classical multi-armed bandit problem as described in Section 3.3. In particular, the UCB policy selects the next arm with the maximum *upper confidence index*, which is defined as the current average reward plus the one-sided confidence interval. As we can see, the upper confidence index can be viewed as an "optimistic" estimate of the reward. However, we note that since we are in a Bayesian setting and our stage-wise rewards are artificially created and thus not *i.i.d.* for each arm, the UCB policy (Auer et al., 2002) cannot be directly applied to our problem.

In fact, our Opt-KG follows a more general principle of "optimism in the face uncertainty" (Szita and Lőrincz, 2008). Essentially, the non-consistency of KG is due to its nature of pure exploitation while a consistent policy should typically utilizes exploration. One of the common techniques to handle the exploration-exploitation dilemma is to take an action based on an optimistic estimation of the rewards (see Szita and Lőrincz, 2008; Even-Dar and Mansour, 2001), which is the role $R^+(a, b)$ plays in Opt-KG.

For our problem, it is also straightforward to design the "pessimistic knowledge gradient" policy which selects the next instance i_t based on the pessimistic outcome of the reward, i.e., $i_t = \arg\max_i \left(R^-(a_i^t, b_i^t) \doteq \min(R_1(a_i^t, b_i^t), R_2(a_i^t, b_i^t))\right)$. However, as shown in the next proposition with the proof in the appendix, the pessimistic KG policy is inconsistent under the uniform prior.

Proposition 8 *When starting from the uniform prior (i.e., $a_i^0 = b_i^0 = 1$) for all θ_i, the pessimistic KG policy will acquire one label for each instance and then consistently acquire the label for the first instance even if the budget T goes to infinity.*

Finally, we discuss some other possible choices of prior distributions. For presentation simplicity, we only consider the Beta prior for each θ_i with the fixed parameters a_i^0 and b_i^0. In practice, more complicated priors can be easily incorporated into our framework. For example, instead of using only one Beta prior, one can adopt a mixture of Beta distributions as the prior and the posterior will also follow a mixture of Beta distributions, which allows an easy inference about the posterior. As we show in the experiments (see Section 8.1.2), the uniform prior does not work well when the data is highly skewed in terms of class distribution. To address this problem, one possible choice is to adopt the prior $p(\theta) = w_1\text{Beta}(c, 1) + w_2\text{Beta}(1, 1) + w_3\text{Beta}(1, c)$ where w_1, w_2 and w_3 are the weights and c is a constant larger than 1 (e.g., $c = 5$). In such a prior, $B(c, 1)$ corresponds to the data with more positive labels while $B(1, c)$ to the data with more negative labels. In addition

to the mixture Beta prior, one can adopt the hierarchical Bayesian approach which puts hyper-priors on the parameters in the Beta priors. The inference can be performed using empirical Bayes approach (Gelman et al., 2013; Robert, 2007). In particular, one can periodically re-calculate the MAP estimate of the hyper-parameters based on the available data and update the model, but otherwise proceed with the given hyper-parameters. For common choices of hyper-priors of Beta, please refer to Section 5.3 in Gelman et al. (2013). These approaches can also be applied to model the workers' reliability as we introduced in the next Section. For example, one can use a mixture of Beta distributions as the prior for the workers' reliability, where $\text{Beta}(c, 1)$ corresponds to reliable workers, $\text{Beta}(1, 1)$ to random workers and $\text{Beta}(1, c)$ to malicious or poorly informed workers.

5. Incorporate Reliability of Heterogeneous Workers

In push crowdsourcing marketplaces, it is important to model workers' reliability so that the decision maker could assign more instances to reliable workers. Assuming that there are M workers in a push marketplace, we can capture the reliability of the j-th worker by introducing an extra parameter $\rho_j \in [0, 1]$ as in (Dawid and Skene, 1979; Raykar et al., 2010; Karger et al., 2013b), which is defined as the probability of getting the same label as the one from a random fully reliable worker. Recall that the soft-label θ_i is the i-th instance's probability of being labeled as positive by a fully reliable worker and let z_{ij} be the label provided by the j-th worker for the i-th instance. We model the distribution of z_{ij} for given θ_i and ρ_j using the *one-coin* model (Dawid and Skene, 1979; Karger et al., 2013b):

$$\begin{aligned}\Pr(z_{ij} = 1|\theta_i, \rho_j) &= \Pr(z_{ij} = 1|y_i = 1, \rho_j)\Pr(y_i = 1|\theta_i) + \Pr(z_{ij} = 1|y_i = -1, \rho_j)\Pr(y_i = -1|\theta_i) \\ &= \rho_j\theta_i + (1 - \rho_j)(1 - \theta_i); \qquad (18) \\ \Pr(z_{ij} = -1|\theta_i, \rho_j) &= \Pr(z_{ij} = -1|y_i = -1, \rho_j)\Pr(y_i = -1|\theta_i) + \Pr(z_{ij} = -1|y_i = 1, \rho_j)\Pr(y_i = 1|\theta_i) \\ &= \rho_j(1 - \theta_i) + (1 - \rho_j)\theta_i, \qquad (19)\end{aligned}$$

where y_i denotes the label provided a random fully reliable worker for the i-th instance. We also note that it is straightforward to extend the current one-coin model to a more complex *two-coin* model (Dawid and Skene, 1979; Raykar et al., 2010) by introducing a pair of parameters (ρ_{j1}, ρ_{j2}) to model the j-th worker's reliability. In particular, ρ_{j1} and ρ_{j2} are the probabilities of getting the positive and negative labels when a fully reliable worker provides the same label.

Here we make the following implicit assumption:

Assumption 9 *We assume that different workers make independent judgments and, for each single worker, the labels provided by him/her to different instances are also independent.*

As the parameter ρ_j increases from 0 to 1, the j-th worker's reliability also increases in the sense that $\Pr(z_{ij} = 1|\theta_i, \rho_j)$ gets more and more close to θ_i, which is the probability of getting a positive label from a random fully reliable worker. Different types of workers can be easily characterized by ρ_j. When all $\rho_j = 1$, it recovers the previous model with fully reliable workers since $\Pr(z_{ij} = 1|\theta_i, \rho_j) = \theta_i$, i.e, each worker provides the label only according to the underlying soft-label of the instance. When $\rho_j = 0.5$, we have $\Pr(z_{ij} = 1|\theta_i, \rho_j) = \Pr(z_{ij} = -1|\theta_i, \rho_j) = 0.5$, which indicates that the j-th worker is a spammer, who randomly submits positive or negative labels. When $\rho_j = 0$, it indicates that the

j-th worker is poorly informed or misunderstands the instruction such that he/she always assigns wrong labels.

We assume that instances' soft-label $\{\theta_i\}_{i=1}^K$ and workers' reliability $\{\rho_j\}_{j=1}^M$ are drawn from known Beta prior distributions: $\theta_i \sim \text{Beta}(a_i^0, b_i^0)$ and $\rho_j \sim \text{Beta}(c_j^0, d_j^0)$. At each stage, we need to make the decision on both the next instance i to be labeled and the next worker j to label the instance i (we omit t in i, j here for notational simplicity). In other words, the action space $\mathcal{A} = \{(i,j) : (i,j) \in \{1,\ldots,K\} \times \{1,\ldots,M\}\}$. Once the decision is made, the distribution of the outcome z_{ij} is given by (18) and (19). Given the prior distributions and likelihood functions in (18) and (19), the Bayesian Markov Decision process can be formally defined as in Section 3. Similar to the homogeneous worker setting, the optimal inferred positive set H_T takes the form of $H_T = \{i : P_i^T \geq 0.5\}$ as in Proposition 2 with $P_i^t = \Pr(i \in H^* | \mathcal{F}_t) = \Pr(\theta_i \geq 0.5 | \mathcal{F}_t)$. The value function $V(S^0)$ still takes the form of (8), which can be further decomposed into the sum of stage-wise rewards in (9) using Proposition 4. Unfortunately, in the heterogeneous worker setting, the posterior distributions of θ_i and ρ_j are highly correlated with a sophisticated joint distribution, which makes the computation of stage-wise rewards in (9) much more challenging. In particular, given the prior $\theta_i \sim \text{Beta}(a_i^0, b_i^0)$ and $\rho_j \sim \text{Beta}(c_j^0, d_j^0)$, the posterior distribution of θ_i and ρ_j given the label $z_{ij} = z \in \{-1, 1\}$ takes the following form:

$$p(\theta_i, \rho_j | z_{ij} = z) = \frac{\Pr(z_{ij} = z | \theta_i, \rho_j)\text{Beta}(a_i^0, b_i^0)\text{Beta}(c_j^0, d_j^0)}{\Pr(z_{ij} = z)}, \tag{20}$$

where $\Pr(z_{ij} = z | \theta_i, \rho_j)$ is the likelihood function defined in (18) and (19) and

$$\begin{aligned}\Pr(z_{ij} = 1) &= \mathbb{E}(\Pr(z_{ij} = 1 | \theta_i, \rho_j)) = \mathbb{E}(\theta_i)\mathbb{E}(\rho_j) + (1 - \mathbb{E}(\theta_i))(1 - \mathbb{E}(\rho_j)) \\ &= \frac{a_i^0}{a_i^0 + b_i^0}\frac{c_j^0}{c_j^0 + d_j^0} + \frac{b_i^0}{a_i^0 + b_i^0}\frac{d_j^0}{c_j^0 + d_j^0}.\end{aligned}$$

As we can see, the posterior distribution $p(\theta_i, \rho_j | z_{ij} = z)$ no longer takes the form of the product of the distributions of θ_i and ρ_j and the marginal posterior of θ_i is no longer a Beta distribution. As a result, P_i^t does not have a simple representation as in (5), which makes the computation of the reward function much more difficult as the number of stages increases. Therefore, to apply our Opt-KG policy to large-scale applications, we need to use some approximate posterior inference techniques.

When applying Opt-KG, we need to perform $2 \cdot K \cdot M \cdot T$ inferences of the posterior distribution in total. Each approximate inference should be computed very efficiently, hopefully in a closed-form. For large-scale problems, most traditional approximate inference techniques such as Markov Chain Monte Carlo (MCMC) or variational Bayesian methods (e.g., Beal, 2003; Paisley et al., 2012) may lead to higher computational cost since each inference is an iterative procedure. To address the computational challenge, we apply the variational approximation with the moment matching technique so that each inference of the approximate posterior can be computed in a closed-form. In fact, any highly efficient approximate inference can be utilized to compute the reward function. Since the main focus of the paper is on the MDP model and Opt-KG policy, we omit the discussion for other possible approximate inference techniques. In particular, we first adopt the variational approximation by

Algorithm 2 Optimistic Knowledge Gradient for Heterogeneous Workers

Input: Parameters of prior distributions for instances $\{a_i^0, b_i^0\}_{i=1}^K$ and for workers $\{c_j^0, d_j^0\}_{j=1}^M$. The total budget T.

for $t = 0, \ldots, T-1$ **do**

1. Select the next instance i_t to label and the next worker j_t to label i_t according to:

$$(i_t, j_t) = \underset{(i,j)\in\{1,\ldots,K\}\times\{1,\ldots,M\}}{\arg\max} \left(R^+(a_i^t, b_i^t, c_j^t, d_j^t) \doteq \max(R_1(a_i^t, b_i^t, c_j^t, d_j^t), R_2(a_i^t, b_i^t, c_j^t, d_j^t))\right). \tag{21}$$

2. Acquire the label $z_{i_t j_t} \in \{-1, 1\}$ of the i-th instance from the j-th worker.

3. Update the posterior by setting:

$$a_{i_t}^{t+1} = \tilde{a}_{i_t}^t(z_{i_t j_t}) \qquad b_{i_t}^{t+1} = \tilde{b}_{i_t}^t(z_{i_t j_t}) \qquad c_{j_t}^{t+1} = \tilde{c}_{j_t}^t(z_{i_t j_t}) \qquad d_{j_t}^{t+1} = \tilde{d}_{j_t}^t(z_{i_t j_t}),$$

and all parameters for $i \neq i_t$ and $j \neq j_t$ remain the same.

end for

Output: The positive set $H_T = \{i : a_i^T \geq b_i^T\}$.

assuming the conditional independence of θ_i and ρ_j:

$$p(\theta_i, \rho_j | z_{ij} = z) \approx p(\theta_i | z_{ij} = z) p(\rho_j | z_{ij} = z).$$

We further approximate $p(\theta_i | z_{ij} = z)$ and $p(\rho_j | z_{ij} = z)$ by two Beta distributions:

$$p(\theta_i | z_{ij} = z) \approx \text{Beta}(\tilde{a}_i(z), \tilde{b}_i(z)), \qquad p(\rho_j | z_{ij} = z) \approx \text{Beta}(\tilde{c}_j(z), \tilde{d}_j(z)),$$

where the parameters $\tilde{a}_i(z)$, $\tilde{b}_i(z)$, $\tilde{c}_j(z)$, $\tilde{d}_j(z)$ are computed using moment matching with the analytical form presented in the appendix. After this approximation, the new posterior distributions of θ_i and ρ_j still have the same structure as their prior distribution, i.e., the product of two Beta distributions, which allows a repeatable use of this approximation every time when a new label is collected. Moreover, due to the Beta distribution approximation of $p(\theta_i | z_{ij} = z)$, the reward function takes a similar form as in the previous setting. In particular, assuming at a certain stage, θ_i has the posterior distribution $\text{Beta}(a_i, b_i)$ and ρ_j has the posterior distribution $\text{Beta}(c_j, d_j)$. The reward of getting positive and negative labels for the i-th instance from the j-th worker are presented in (22) and (23):

$$R_1(a_i, b_i, c_j, d_j) = h(I(\tilde{a}_i(z=1), \tilde{b}_i(z=1))) - h(I(a_i, b_i)), \tag{22}$$

$$R_2(a_i, b_i, c_j, d_j) = h(I(\tilde{a}_i(z=-1), \tilde{b}_i(z=-1))) - h(I(a_i, b_i)). \tag{23}$$

With the reward in place, we present Opt-KG for budget allocation in the heterogeneous worker setting in Algorithm 2. We also note that due to the variational approximation of the posterior, establishing the consistency results of Opt-KG becomes very challenging in the heterogeneous worker setting.

6. Extensions

Our MDP formulation is a general framework to address many complex settings of dynamic budget allocation problems in crowd labeling. In this section, we briefly discuss two

important extensions, where for both extensions, Opt-KG can be directly applied as an approximate policy. We note that for the sake of presentation simplicity, we only present these extensions in the noiseless homogeneous worker setting. Further extensions to the heterogeneous setting are rather straightforward using the technique from Section 5.

6.1 Utilizing Contextual Information

When the contextual information is available for instances, we could easily extend our model to incorporate such an important information. In particular, let the contextual information for the i-th instance be represented by a p-dimensional feature vector $\mathbf{x}_i \in \mathbb{R}^p$. We could utilize the feature information by assuming a logistic model for θ_i:

$$\theta_i \doteq \frac{\exp\{\langle \mathbf{w}, \mathbf{x}_i \rangle\}}{1 + \exp\{\langle \mathbf{w}, \mathbf{x}_i \rangle\}},$$

where $\mathbf{w}$ is assumed to be drawn from a Gaussian prior $N(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$. At the t-th stage with the current state $(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$, the decision maker determines the instance i_t and acquire its label $y_{i_t} \in \{-1, 1\}$. Then we update the posterior $\boldsymbol{\mu}_{t+1}$ and $\boldsymbol{\Sigma}_{t+1}$ using the Laplace method as in Bayesian logistic regression (Bishop, 2007). Variational methods can be applied to further accelerate the posterior update (Jaakkola and Jordan, 2000). The details are provided in the appendix.

6.2 Multi-Class Categorization

Our MDP formulation can also be extended to deal with multi-class categorization problems, where each instance is a multiple choice question with several possible options (i.e., classes). More formally, in a multi-class setting with C different classes, we assume that the i-th instance is associated with a probability vector $\boldsymbol{\theta}_i = (\theta_{i1}, \ldots \theta_{iC})$, where θ_{ic} is the probability that the i-th instance will be labeled as the class c by a random fully reliable worker and $\sum_{i=1}^{C} \theta_{ic} = 1$. We assume that $\boldsymbol{\theta}_i$ has a Dirichlet prior $\boldsymbol{\theta}_i \sim \text{Dir}(\boldsymbol{\alpha}_i^0)$ and the initial state S^0 is a $K \times C$ matrix with $\boldsymbol{\alpha}_i^0$ as its i-th row. At each stage t with the current state S^t, we determine the next instance i_t to be labeled and collect its label $y_{i_t} \in \{1, \ldots, C\}$, which follows the categorical distribution: $p(y_{i_t}) = \prod_{c=1}^{C} \theta_{i_t c}^{I(y_{i_t}=c)}$. Since the Dirichlet is the conjugate prior of the categorical distribution, the next state induced by the posterior distribution is: $S_{i_t}^{t+1} = S_{i_t}^t + \boldsymbol{\delta}_{y_{i_t}}$ and $S_i^{t+1} = S_i^t$ for all $i \neq i_t$. Here $\boldsymbol{\delta}_c$ is a row vector with one at the c-th entry and zeros at all other entries. The transition probability is:

$$\Pr(y_{i_t} = c | S^t, i_t) = \mathbb{E}(\theta_{i_t c} | S^t) = \frac{\alpha_{i_t c}^t}{\sum_{c=1}^{C} \alpha_{i_t c}^t}.$$

We denote the true set of instances in class c by $H_c^* = \{i : \theta_{ic} \geq \theta_{ic'}, \forall c' \neq c\}$. By a similar argument as in Proposition 2, at the final stage T, the estimated set of instances belonging to class c is

$$H_c^T = \{i : P_{ic}^T \geq P_{ic'}^T, \forall c' \neq c\},$$

where $P_{ic}^t = \Pr(i \in H_c^* | \mathcal{F}_t) = \Pr(\theta_{ic} \geq \theta_{ic'}, \quad \forall \quad c' \neq c | S^t)$. We note that if the i-th instance belongs to more than one H_c^T, we only assign it to the one with the smallest

index c so that $\{H_c^T\}_{c=1}^C$ forms a partition of $\{1,\ldots,K\}$. Let $\mathbf{P}_i^t = (P_{i1}^t,\ldots,P_{iC}^t)$ and $h(\mathbf{P}_i^t) = \max_{1\le c\le C} P_{ic}^t$. The expected reward takes the form of:

$$R(S^t, i_t) = \mathbb{E}\left(h(\mathbf{P}_{i_t}^{t+1}) - h(\mathbf{P}_{i_t}^t)|S^t, i_t\right).$$

With the reward function in place, we can formulate the problem into a MDP and use DP to obtain the optimal policy and Opt-KG to compute an approximate policy. The only computational challenge is how to calculate P_{ic}^t efficiently so that the reward can be evaluated. We present an efficient method in the appendix. We can further use Dirichlet distribution to model workers reliability as in Liu and Wang (2012). Using multi-class Bayesian logistic regression, we can also incorporate contextual information into the multi-class setting in a straightforward manner.

7. Related Works

Categorical crowd labeling is one of the most popular tasks in crowdsourcing since it requires less effort of the workers to provide categorical labels than other tasks such as language translations. Most work in categorical crowd labeling are solving a static problem, i.e., inferring true labels and workers' reliability based on a static labeled data set (Dawid and Skene, 1979; Raykar et al., 2010; Liu and Wang, 2012; Welinder et al., 2010; Whitehill et al., 2009; Zhou et al., 2012; Liu et al., 2012; Gao and Zhou, 2013). The first work that incorporates diversity of worker reliability is by Dawid and Skene (1979), which uses EM to perform the point estimation on both worker reliability and true class labels. Based on that, Raykar et al. (2010) extended (Dawid and Skene, 1979) by introducing Beta prior for workers' reliability and features of instances in the binary setting; and Liu and Wang (2012) further introduced Dirichlet prior for modeling workers' reliability in the multi-class setting. Our work utilizes the modeling techniques in these two static models as basic building blocks but extends to dynamic budget allocation settings.

In recent years, there are several works that have been devoted into online learning or budget allocation in crowdsourcing (Karger et al., 2013a,b; Bachrach et al., 2012; Ho et al., 2013; Ertekin et al., 2012; Yan et al., 2011; Kamar et al., 2012; Ipeirotis et al., 2013). The method proposed in Karger et al. (2013b) is based on the one-coin model. In particular, it assigns instances to workers according to a random regular bipartite graph. Although the error rate is proved to achieve the minimax rate, its analysis is asymptotic and method is not optimal when the budget is limited. Karger et al. (2013a) further extended the work by Karger et al. (2013b) to the multi-class setting. The new labeling uncertainty method in Ipeirotis et al. (2013) is one of the state-of-the-art methods for repeated labeling. However, it does not model each worker's reliability and incorporate it into the allocation process. Ho et al. (2013) proposed an online primal dual method for adaptive task assignment and investigated the sample complexity to guarantee that the probability of making an error for each instance is less that a threshold. However, it requires gold samples to estimate workers' reliability. Kamar et al. (2012) used MDP to address a different decision problem in crowd labeling, where the decision maker collects labels for each instance one after another and only decides whether to hire an additional worker or not. Basically, it is an optimal stopping problem since there is no pre-fixed amount of budget and one needs to balance the accuracy v.s. the amount of budget. Since the accuracy and the amount of budget are in different

metrics, such a balance could be very subjective. Furthermore, the MDP framework in Kamar et al. (2012) cannot distinguish different workers. To the best of our knowledge, there is no existing method that characterizes the *optimal* allocation policy for finite T. In this work, with the MDP formulation and DP algorithm, we characterize the optimal policy for budget allocation in crowd labeling under any budget level.

We also note that the budget allocation in crowd labeling is fundamentally different from noisy active learning (Settles, 2009; Nowak, 2009). Active learning usually does not model the variability of labeling difficulties among instances and assumes a single (noisy) oracle; while in crowd labeling, we need to model both instances' labeling difficulty and different workers' reliability. Secondly, active learning requires the feature information of instances for the decision, which could be unavailable in crowd labeling. Finally, the goal of the active learning is to label as few instances as possible to learn a good classifier. In contrast, for budget allocation in crowd labeling, the goal is to infer the true labels for as many instances as possible.

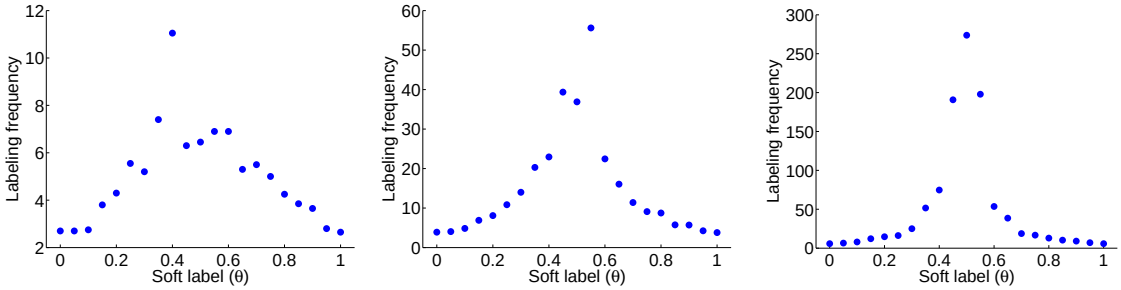
In fact, our MDP formulation is essentially a finite-horizon Bayesian multi-armed bandit (MAB) problem. While the infinite-horizon Bayesian MAB has been well-studied and the optimal policy can be computed via Gittins index (Gittins, 1989), for finite-horizon Bayesian MAB, the Gittins index rule is only an approximate policy with high computational cost. The proposed Opt-KG and a more general conditional value-at-risk based KG could be general policies for Bayesian MAB. Recently, a Bayesian UCB policy was proposed to address a different Bayesian MAB problem (Kaufmann et al., 2012). However, it is not clear how to directly apply the policy to our problem since we are not updating the posterior of the mean of rewards as in Kaufmann et al. (2012). We note that our problem is also related to optimal stopping problem. The main difference is that the optimal stopping problem is infinite-horizon while our problem is finite-horizon and the decision process must stop when the budget is exhausted.

8. Experiments

In this section, we conduct empirical study to show some interesting properties of the proposed Opt-KG policy and compare its performance to other methods. We observe that several commonly used priors such as the uniform prior (Beta$(1,1)$), Jeffery prior (Beta$(1/2,1/2)$) and Haldane prior (Beta$(0,0)$) for instances' soft-label $\{\theta_i\}_{i=1}^K$ lead to very similar performance. Therefore, we adopt the uniform prior (Beta$(1,1)$) unless otherwise specified. In addition, for the ease of comparison, the accuracy is defined as $\frac{|H_T \cap H^*| + |(H_T)^c \cap (H^*)^c|}{K}$, which is normalized between $[0,1]$.

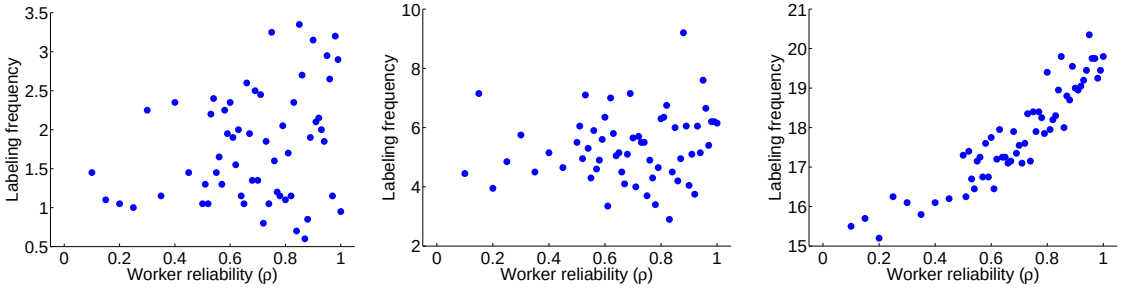
8.1 Simulation Study

In this section, we conduct simulated study. For each simulated experiment, we randomly generate 20 different sets of data and report the averaged accuracy. The deviations for different methods are similar and quite small and thus omitted for the purpose of better visualization and space-saving.



(a) $T = 5K = 105$ (b) $T = 15K = 315$ (c) $T = 50K = 1050$

Figure 4: Labeling counts for instances with different levels of ambiguity.

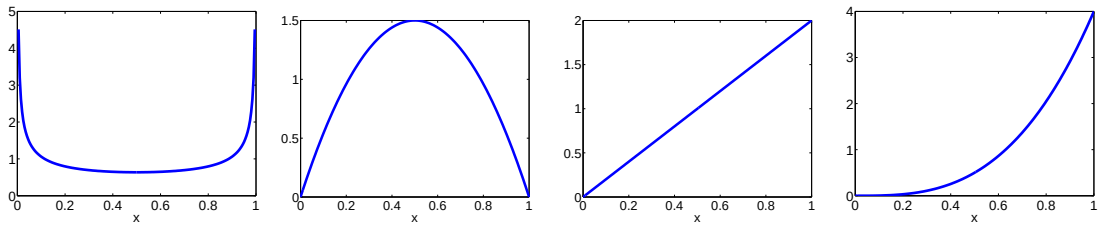


(a) $T = 5K = 105$ (b) $T = 15K = 315$ (c) $T = 50K = 1050$

Figure 5: Labeling counts for workers with different levels of reliability.

8.1.1 Study on Labeling Frequency

We first investigate that, in the homogeneous noiseless worker setting (i.e., workers are fully reliable), how the total budget is allocated among instances with different levels of ambiguity. In particular, we assume there are $K = 21$ instances with soft-labels $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3, \ldots, \theta_K) = (0, 0.05, 0.1, \ldots, 1)$. We vary the total budget $T = 5K, 15K, 50K$ and report the number of times that each instance is labeled on average over 20 independent runs. The results are presented in Figure 4. It can be seen from Figure 4 that, more ambiguous instances with θ close to 0.5 in general receive more labels than those simple instances with θ close to 0 or 1. A more interesting observation is that when the budget level is low (e.g., $T = 5K$ in Figure 4(a)), the policy spends less budget on those very ambiguous instances (e.g., $\theta = 0.45$ or 0.5), but more budget on exploring less ambiguous instances (e.g., $\theta = 0.35$, 0.4 or 0.6). When the budget goes higher (e.g., $T = 15K$ in Figure 4(b)), those very ambiguous instances receive more labels but the most ambiguous instance ($\theta = 0.5$) not necessarily receives the most labels. In fact, the instances with $\theta = 0.45$ and $\theta = 0.55$ receive more labels than that of the most ambiguous instance. When the total budget is sufficiently large (e.g., $T = 50K$ in Figure 4(c)), the most ambiguous instance receives the most labels since all the other instances have received enough labels to infer their true labels.



(a) Beta$(0.5, 0.5)$ (b) Beta$(2, 2)$ (c) Beta$(2, 1)$ (d) Beta$(4, 1)$

Figure 6: Density plot for different Beta distributions for generating each θ_i. Here, (a) represents there are more easier instances; (b) more ambiguous instances; (c) & (d) imbalanced class distributions with more positive instances.

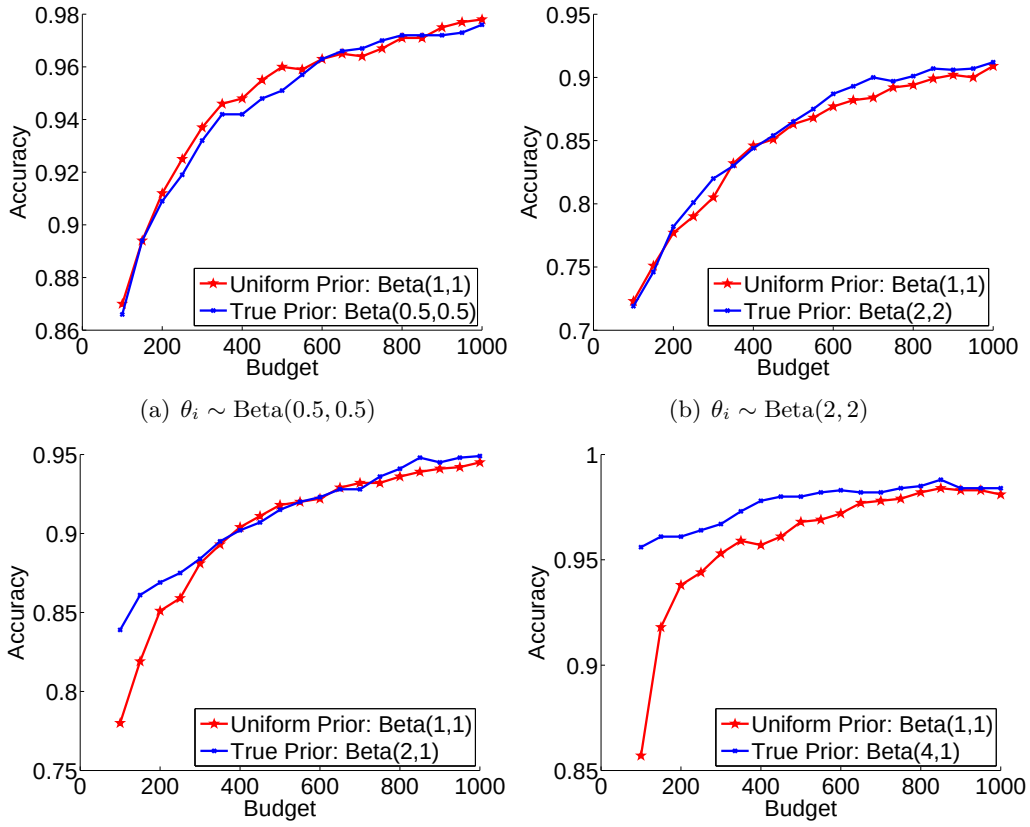
Next, we investigate that, in the heterogeneous worker setting, how many instances each worker is assigned. We simulate $K = 21$ instances' soft-labels as before and further simulate workers' reliability $\boldsymbol{\rho} = (\rho_1, \rho_2, \ldots, \rho_M) = (0.1, 0.15, \ldots, 0.5, 0.505, 0.515, \ldots, 0.995)$ for $M = 59$ workers. Such a simulation ensures that there are more reliable workers, which is in line with actual situation. We vary the total budget $T = 5K, 15K, 50K$ and report the number of instances that each worker is assigned on average over 20 independent runs in Figure 5. As one can see, when the budget level goes up, there is clear trend that more reliable workers receive more instances.

8.1.2 Prior for Instances

We investigate how robust Opt-KG is when using the uniform prior for each θ_i. We first simulate $K = 50$ instances with each $\theta_i \sim$ Beta$(0.5, 0.5)$, $\theta_i \sim$ Beta$(2, 2)$, $\theta_i \sim$ Beta$(2, 1)$ or $\theta_i \sim$ Beta$(4, 1)$. The density functions of these four different Beta distributions are plotted in Figure 6. For each generating distribution of θ_i, we compare Opt-KG using the uniform prior (Beta$(1, 1)$) (in red line) to Opt-KG with the true generating distribution as the prior (in blue line). The comparison in accuracy with different levels of budget ($T = 2K, \ldots, 20K$) is shown in Figure 7. As we can see, the performance of Opt-KG using two different priors are quite similar for most generating distributions except for $\theta_i \sim$ Beta$(4, 1)$ (i.e., the highly imbalanced class distribution). When $\theta_i \sim$ Beta$(4, 1)$, the Opt-KG with uniform prior needs at least $T = 16K$ units of budget to match the performance of Opt-KG with true generating distribution as the prior. This result indicates that for balanced class distributions, the uniform prior is a good choice and robust to the underlying distribution of θ_i. For highly imbalanced class distributions, if a uniform prior is adopted, one needs more budget to recover from the inaccurate prior belief.

8.1.3 Prior on Workers

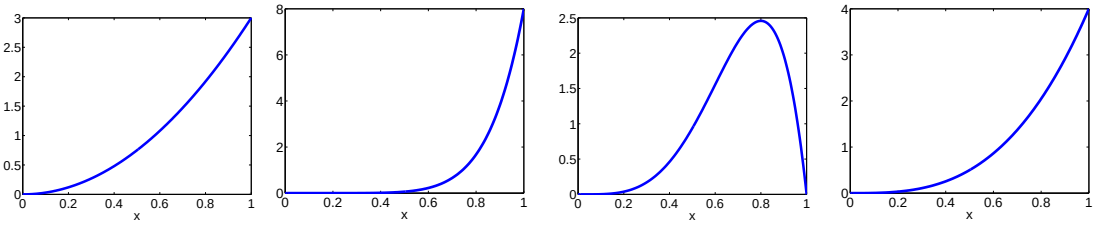
We investigate how sensitive the prior for the workers' reliability ρ_j is. In particular, we simulate $K = 50$ instances with each $\theta_i \sim$ Beta$(1, 1)$ and $M = 100$ workers with $\rho_j \sim$ Beta$(3, 1)$, $\rho_j \sim$ Beta$(8, 1)$ or $\rho_j \sim$ Beta$(5, 2)$. We ensure that there are more reliable workers than spammers or poorly informed workers, which is in line with the actual situation. We use the



(a) $\theta_i \sim \text{Beta}(0.5, 0.5)$ (b) $\theta_i \sim \text{Beta}(2, 2)$

(c) $\theta_i \sim \text{Beta}(2, 1)$ (d) $\theta_i \sim \text{Beta}(4, 1)$

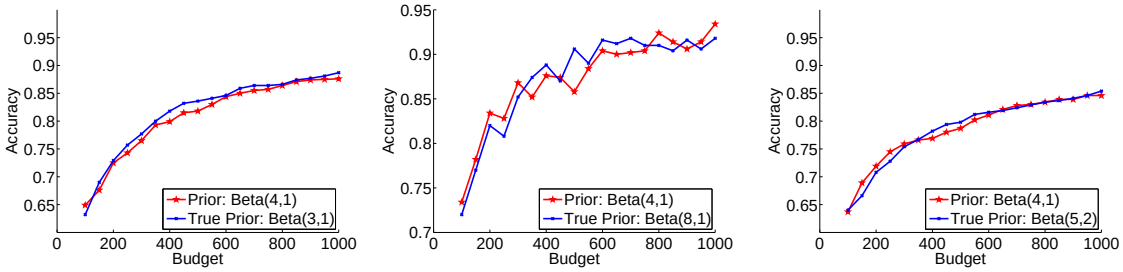
Figure 7: Comparison between Opt-KG using the uniform distribution and true generating distribution as the prior.



(a) Beta$(3, 1)$ (b) Beta$(8, 1)$ (c) Beta$(5, 2)$ (d) Beta$(4, 1)$

Figure 8: Density plot for different Beta distributions for generating ρ_j. The plot in (d) is the one that we use as the prior.

prior Beta$(4, 1)$, which indicates that we have the prior belief that most workers preform reasonably well and the averaged accuracy is $4/5 = 80\%$. In Figure 8, we show different density functions for generating ρ_j and the prior that we use (in Figure 8 (d)). For each



(a) $\theta_i \sim \text{Beta}(3,1)$ (b) $\theta_i \sim \text{Beta}(8,1)$ (c) $\theta_i \sim \text{Beta}(5,2)$

Figure 9: Comparison between Opt-KG using $\text{Beta}(4,1)$ and true generating distribution prior as the prior.

generating distribution of θ_i, we compare the Opt-KG policy using the prior ($\text{Beta}(4,1)$) (in red line) to the Opt-KG with the true generating distribution as the prior (in blue line). The comparison in accuracy with different levels of budget ($T = 2K, \ldots, 20K$) is shown in Figure 9. From Figure 9, we observe that the performance of Opt-KG using two different priors are quite similar in all different settings. Hence, we will use $\text{Beta}(4,1)$ as the prior when the true prior of workers is unavailable.

8.1.4 Performance Comparison Under the Homogeneous Noiseless Worker Setting

We compare the performance of Opt-KG under the homogeneous noiseless worker setting to several other competitors, including

1. Uniform: Uniform sampling.
2. KG(Random): Randomized knowledge gradient (Frazier et al., 2008).
3. Gittins-Inf: A Gittins-indexed based policy proposed by Xie and Frazier (2013) for solving an infinite-horizon Bayesian MAB problem where the reward is discounted by δ. Although it solves a different problem, we apply it as a heuristic by choosing the discount factor δ such that $T = 1/(1-\delta)$.
4. NLU: The "new labeling uncertainty" method proposed by Ipeirotis et al. (2013).

We note that we do not compare to the finite-horizon Gittins index rule (Nino-Mora, 2011) since its computation is very expensive. On some small-scale problems, we observe that the finite-horizon Gittins index rule (Nino-Mora, 2011) has the similar performance as Gittins-Inf in Xie and Frazier (2013).

We simulate $K = 50$ instances with each $\theta_i \sim \text{Beta}(1,1)$, $\theta_i \sim \text{Beta}(0.5,0.5)$, $\theta_i \sim \text{Beta}(2,2)$, $\theta_i \sim \text{Beta}(2,1)$ or $\theta_i \sim \text{Beta}(4,1)$ (see Figure 6). For each of the five settings, we vary the total budget $T = 2K, 3K, \ldots, 20K$ and report the mean of accuracy for 20 independently generated sets of $\{\theta_i\}_{i=1}^K$. For the last four settings, we report the comparison among different methods when either using the uniform prior ("uni prior" for short) or the

true generating distribution as the prior. From Figure 10, the proposed Opt-KG outperforms all the other competitors in most settings regardless the choice of the prior. For $\theta_i \sim \text{Beta}(0.5, 0.5)$, NLU matches the performance of Opt-KG; and for $\theta_i \sim \text{Beta}(2, 2)$, Gittins-inf matches the performance of Opt-KG. We also observe that the performance of randomized KG only slightly improves that of uniform sampling.

8.1.5 Performance Comparison Under the Heterogeneous Worker Setting

We compare the proposed Opt-KG under the heterogeneous worker setting to several other competitors:

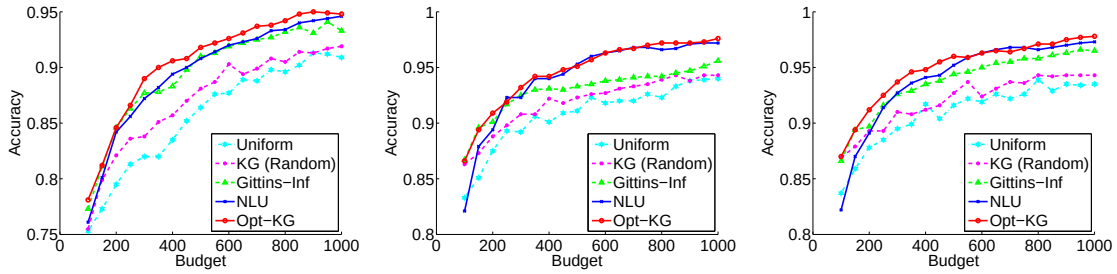
1. Uniform: Uniform sampling.
2. KG(Random): Randomized knowledge gradient (Frazier et al., 2008).
3. KOS: The randomized budget allocation algorithm in Karger et al. (2013b).

We note that several competitors for the homogeneous worker setting (e.g., Gittins-inf and NLU) cannot be directly applied to the heterogeneous worker setting since they fail to model each worker's reliability.

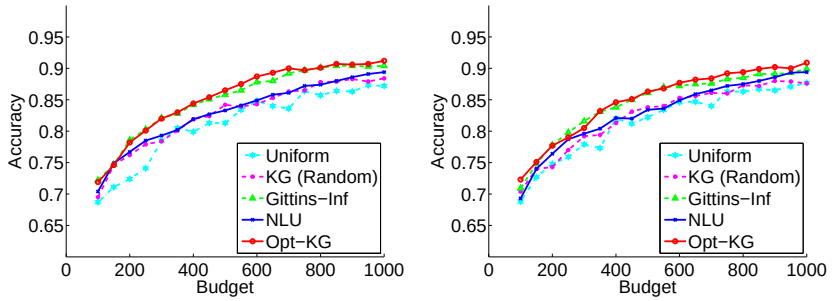
We simulate $K = 50$ instances with each $\theta_i \sim \text{Beta}(1, 1)$ and $M = 100$ workers with $\rho_j \sim \text{Beta}(4, 1)$, $\rho_j \sim \text{Beta}(3, 1)$, $\rho_j \sim \text{Beta}(8, 1)$ or $\rho_j \sim \text{Beta}(5, 2)$ (see Figure 8). For each of the four settings, we vary the total budget $T = 2K, 3K, \ldots, 20K$ and report the mean of accuracy for 20 independently generated sets of parameters. For the last three settings, we report the comparison among different methods when either using $\text{Beta}(4, 1)$ prior or the true generating distribution for ρ_j as the prior. From Figure 11, the proposed Opt-KG outperforms all the other competitors regardless the choice of the prior.

8.2 Real Data

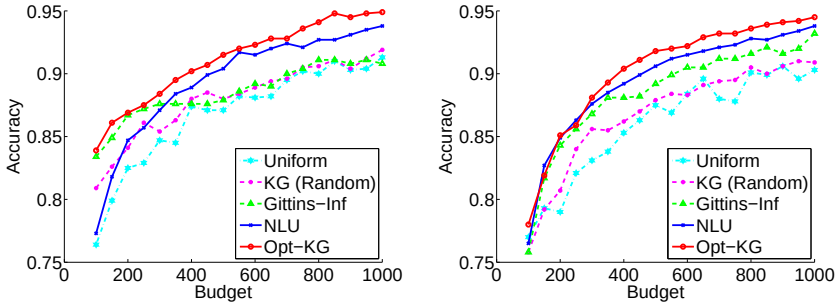
We compare different policies on a standard real data set for recognizing textual entailment (RTE) (Section 4.3 in Snow et al., 2008). There are 800 instances and each instance is a sentence pair. Each sentence pair is presented to 10 different workers to acquire binary choices of whether the second hypothesis sentence can be inferred from the first one. There are in total 164 different workers. We first consider the homogeneous noiseless setting without incorporating the diversity of workers and use the uniform prior ($\text{Beta}(1, 1)$) for each θ_i. In such a setting, once we decide to label an instance, we randomly choose a worker (who provides the label in the full data set) to acquire the label. Due to this randomness, we run each policy 20 times and report the mean of the accuracy in Figure 12(a). As we can see, Opt-KG, Gittins-inf and NLU all perform quite well. We also note that although Gittins-inf performs slightly better than our method on this data, it requires solving a linear system with $O(T^2)$ variables at each stage, which could be too expensive for large-scale applications. While our Opt-KG policy has a time complexity linear in KT and space complexity linear in K, which is much more efficient when a quick online decision is required. In particular, we present the comparison between Opt-KG and Gittins-inf on the averaged CPU time under different budget levels in Table 4. As one can see, Gittins-inf is computationally more expensive than Opt-KG.



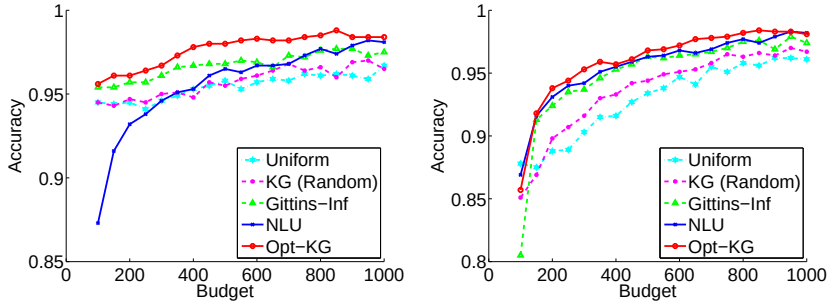
(a) $\theta_i \sim \text{Beta}(1,1)$ (True Prior) (b) $\theta_i \sim \text{Beta}(0.5,0.5)$ (True Prior) (c) $\theta_i \sim \text{Beta}(0.5,0.5)$ (Uni Prior)



(d) $\theta_i \sim \text{Beta}(2,2)$ (True Prior) (e) $\theta_i \sim \text{Beta}(2,2)$ (Uni Prior)



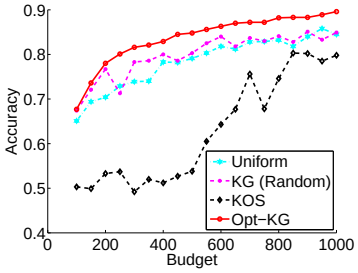
(f) $\theta_i \sim \text{Beta}(2,1)$ (True Prior) (g) $\theta_i \sim \text{Beta}(2,1)$ (Uni Prior)



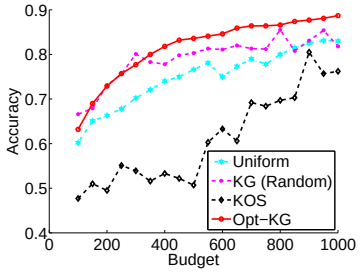
(h) $\theta_i \sim \text{Beta}(4,1)$ (True Prior) (i) $\theta_i \sim \text{Beta}(4,1)$ (Uni Prior)

Figure 10: Performance comparison under the homogeneous noiseless worker setting.

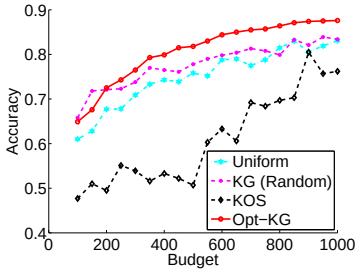
When the worker reliability is incorporated, we compare different policies in Figure 12(b). We put a $\text{Beta}(4,1)$ prior distribution for each ρ_j which indicates that we have the



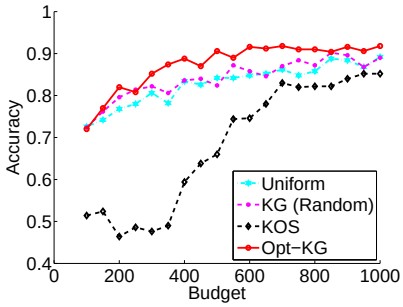
(a) $\rho_j \sim \text{Beta}(4,1)$ (True Prior)



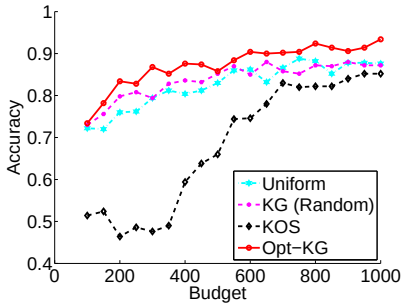
(b) $\rho_j \sim \text{Beta}(3,1)$ (True Prior)



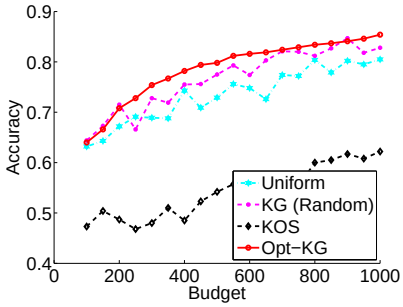
(c) $\rho_j \sim \text{Beta}(3,1)$ (Beta(4, 1) Prior)



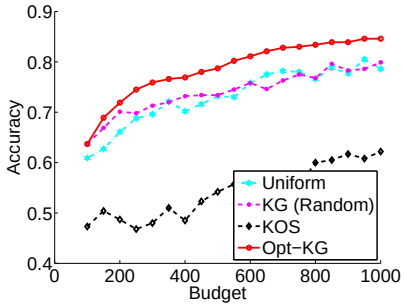
(d) $\rho_j \sim \text{Beta}(8,1)$ (True Prior)



(e) $\rho_j \sim \text{Beta}(8,1)$ (Beta(4, 1) Prior)



(f) $\rho_j \sim \text{Beta}(5,2)$ (True Prior)



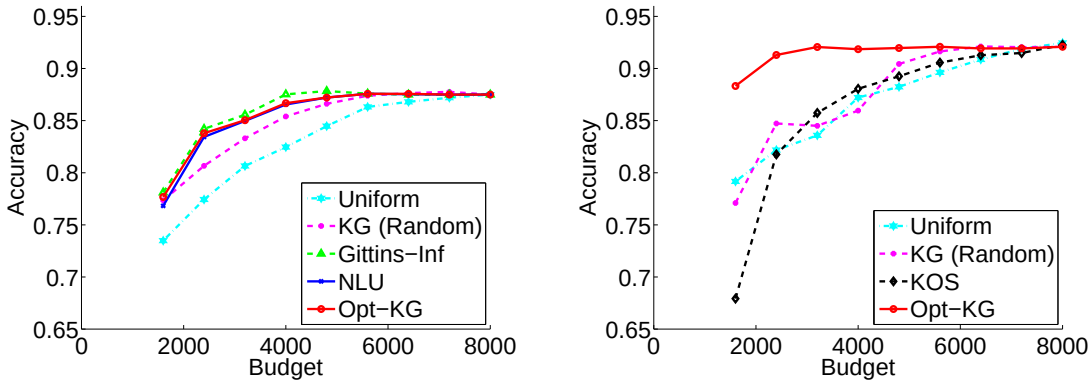
(g) $\rho_j \sim \text{Beta}(5,2)$ (Beta(4, 1) Prior)

Figure 11: Performance comparison under the heterogeneous worker setting.

Budget T	$2K = 1,600$	$4K = 3,200$	$6K = 4,800$	$10K = 8,000$
Opt-KG	1.09	2.19	3.29	5.48
Gittins-inf	25.87	35.70	45.59	130.68

Table 4: Comparison in CPU time (seconds)

prior belief that most workers perform reasonably well. Other priors in Figure 8 lead to similar results and thus omitted here. As one can see, the accuracy of Opt-KG is much higher than that of other policies when T is small. It achieves the highest accuracy of 92.05% only using 40% of the total budget (i.e., on average, each instance is labeled 4 times). One may also observe that when $T > 4K = 3,200$, the performance of Opt-KG does not improve



(a) RTE: Homogeneous Noiseless Worker

(b) RTE: Heterogeneous Worker

Figure 12: Performance comparison on the real data set.

and in fact, slightly downgrades a little bit. This is mainly due to the restrictiveness of the experimental setting. In particular, since the experiment is conducted on a fixed data set with partially observed labels, the Opt-KG cannot freely choose instance-worker pairs especially when the budget goes up (i.e., the action set is greatly restricted). According to our experience, such a phenomenon will not happen on experiments when labels can be obtained from any instance-worker pair. Comparing Figure 12(b) to 12(a), we also observe that Opt-KG under the heterogeneous worker setting performs much better than Opt-KG under the homogeneous worker setting, which indicates that it is beneficial to incorporate workers' reliability.

9. Conclusions and Future Work

In this paper, we propose to address the problem of budget allocation in crowd labeling. We model the problem using the Bayesian Markov decision process and characterize the optimal policy using the dynamic programming. We further propose a computationally more attractive approximate policy: optimistic knowledge gradient. Our MDP formulation is a general framework, which can be applied to binary or multi-class, contextual or non-contextual crowd labeling problems in either pull or push crowdsourcing marketplaces.

There are several possible future directions for this work. First, it is of great interest to show the consistency of Opt-KG in heterogeneous worker setting and further provide the theoretical results on the performance of Opt-KG under finite budget. Second, in this work, we assume that both instances and workers are equally priced. Although this assumption is standard in many crowd labeling applications, a dynamic pricing strategy as the allocation process proceeds will better motivate those more reliable workers to label more challenge instances. A recent work in Wang et al. (2013) provides some quality-based pricing algorithms for crowd workers and it will be interesting to incorporate their strategies into our dynamic allocation framework. Third, we assume that the labels provided by the same worker to different instances are independent. It is more interesting to consider that the workers' reliability will be improved during the labeling process when some useful feedback

can be provided. Further, since the proposed Opt-KG is a fairly general approximate policy for MDP, it is also interesting to apply it to other statistical decision problems.

Acknowledgments

We would like to thank Qiang Liu for sharing the code for KOS method; Jing Xie and Peter Frazier for sharing their code for computing infinite-horizon Gittins index; John Platt, Chris Burges and Kevin Murphy for helpful discussions; and anonymous reviewers and the associate editor for their constructive comments on improving the quality of the paper.

Appendix A. Proof of Results

In this section, we provide the proofs of the main results of our paper.

A.1 Proof of Proposition 2

The final positive set H_T is chosen to maximize the expected accuracy conditioned on $\mathcal{F}_T$:

$$H_T = \arg\max_H \mathbb{E}\left(\sum_{i=1}^{K} \left(\mathbf{1}(i \in H)\mathbf{1}(i \in H^*) + \mathbf{1}(i \notin H)\mathbf{1}(i \notin H^*)\right) \middle| \mathcal{F}_T\right). \tag{24}$$

According to the definition (6) of P_i^T, we can re-write (24) using the linearity of the expectation:

$$\begin{aligned} & \sum_{i=1}^{K} \left(\mathbf{1}(i \in H)\Pr(i \in H^*|\mathcal{F}_T) + \mathbf{1}(i \notin H)\Pr(i \notin H^*|\mathcal{F}_T)\right) \\ = & \sum_{i=1}^{K} \left(\mathbf{1}(i \in H)P_i^T + \mathbf{1}(i \notin H)(1 - P_i^T)\right). \end{aligned} \tag{25}$$

To maximize (25) over H, it easy to see that we should set $i \in H$ if and only if $P_i^T \geq 0.5$. Therefore, we have the positive set

$$H_T = \{i : P_i^T \geq 0.5\}.$$

A.2 Proof of Corollary 3

Recall that

$$I(a,b) = \Pr(\theta \geq 0.5|\theta \sim \text{Beta}(a,b)) = \frac{1}{B(a,b)} \int_{0.5}^{1} t^{a-1}(1-t)^{b-1}\mathrm{d}t, \tag{26}$$

where $B(a,b)$ is the beta function.

It is easy to see that $I(a,b) > 0.5 \Longleftrightarrow I(a,b) > 1 - I(a,b)$. We re-write $1 - I(a,b)$ as follows

$$1 - I(a,b) = \frac{1}{B(a,b)} \int_{0}^{0.5} t^{a-1}(1-t)^{b-1}\mathrm{d}t = \frac{1}{B(a,b)} \int_{0.5}^{1} t^{b-1}(1-t)^{a-1}\mathrm{d}t,$$

where the second equality is obtained by setting $t := 1 - t$. Then we have:

$$\begin{aligned} I(a,b) - (1 - I(a,b)) &= \frac{1}{B(a,b)} \int_{0.5}^{1} (t^{a-1}(1-t)^{b-1} - t^{b-1}(1-t)^{a-1}) \mathrm{d}t \\ &= \frac{1}{B(a,b)} \int_{0.5}^{1} t^{a-1}(1-t)^{b-1} \left(\left(\frac{t}{1-t} \right)^{a-b} - 1 \right) \mathrm{d}t. \end{aligned}$$

Since $t > 0.5$, $\frac{t}{1-t} > 1$. When $a > b$, $\left(\frac{t}{1-t}\right)^{a-b} > 1$ and hence $I(a,b) - (1 - I(a,b)) > 0$, i.e, $I(a,b) > 0.5$. When $a = b$, $\left(\frac{t}{1-t}\right)^{a-b} \equiv 1$ and $I(a,b) = 0.5$. When $a < b$, $\left(\frac{t}{1-t}\right)^{a-b} < 1$ and $I(a,b) < 0.5$.

A.3 Proof of Proposition 4

We use the proof technique in Xie and Frazier (2013) to prove Proposition 4. According to (8), the value function takes the following form,

$$V(S^0) = \sup_{\pi} \mathbb{E}^{\pi} \left(\sum_{i=1}^{K} h(P_i^T) \right). \tag{27}$$

To decompose the final accuracy $\sum_{i=1}^{K} h(P_i^T)$ into the incremental reward at each stage, we define $G_0 = \sum_{i=1}^{K} h(P_i^0)$ and $G_{t+1} = \sum_{i=1}^{K} h(P_i^{t+1}) - \sum_{i=1}^{K} h(P_i^t)$. Then, $\sum_{i=1}^{K} h(P_i^T)$ can be decomposed as: $\sum_{i=1}^{K} h(P_i^T) \equiv G_0 + \sum_{t=0}^{T-1} G_{t+1}$. The value function can now be re-written as follows:

$$\begin{aligned} V(S^0) &= G_0(S^0) + \sup_{\pi} \sum_{t=0}^{T-1} \mathbb{E}^{\pi}(G_{t+1}) \\ &= G_0(S^0) + \sup_{\pi} \sum_{t=0}^{T-1} \mathbb{E}^{\pi} \left(\mathbb{E}(G_{t+1} | \mathcal{F}_t) \right) \\ &= G_0(S^0) + \sup_{\pi} \sum_{t=0}^{T-1} \mathbb{E}^{\pi} \left(\mathbb{E}(G_{t+1} | S^t, i_t) \right). \end{aligned}$$

Here, the first inequality is true because G_0 is determinant and independent of π; the second inequality is due to the tower property of conditional expectation and the third one holds because G_{t+1}, which is a function of P_i^{t+1} and P_i^t, depends on $\mathcal{F}_t$ only through S^t and i_t. We define incremental expected reward gained by labeling the i_t-th instance at the state S^t as follows:

$$\begin{aligned} R(S^t, i_t) &= \mathbb{E}(G_{t+1} | S^t, i_t) = \mathbb{E} \left(\sum_{i=1}^{K} h(P_i^{t+1}) - \sum_{i=1}^{K} h(P_i^t) | S^t, i_t \right) \\ &= \mathbb{E} \left(h(P_{i_t}^{t+1}) - h(P_{i_t}^t) | S^t, i_t \right). \end{aligned} \tag{28}$$

The last equation is due to the fact that only $P_{i_t}^t$ will be changed if the i_t-th instance is labeled next. With the expected reward function in place, the value function in (8) can be re-formulated as:

$$V(S^0) = G_0(\mathbf{s}) + \sup_{\pi} \mathbb{E}^{\pi} \left(\sum_{t=0}^{T-1} R(S^t, i_t) \Big| S^0 \right). \tag{29}$$

A.4 Proof of Proposition 6

To prove the failure of deterministic KG, we first show a key property for the expected reward function:

$$R(a,b) = \frac{a}{a+b}\left(h(I(a+1,b)) - h(I(a,b))\right) + \frac{b}{a+b}\left(h(I(a,b+1)) - h(I(a,b))\right). \tag{30}$$

Lemma 10 *When a, b are positive integers, if $a = b$, $R(a,b) = \frac{0.5^{2a}}{aB(a,a)}$ and if $a \neq b$, $R(a,b) = 0$.*

To prove lemma 10, we first present several basic properties for $B(a,b)$ and $I(a,b)$, which will be used in all the following theorems and proofs.

1. Properties for $B(a,b)$:

$$B(a,b) = B(b,a), \tag{31}$$
$$B(a+1,b) = \frac{a}{a+b}B(a,b), \tag{32}$$
$$B(a,b+1) = \frac{b}{a+b}B(a,b). \tag{33}$$

2. Properties for $B(a,b)$:

$$I(a,b) = 1 - I(b,a), \tag{34}$$
$$I(a+1,b) = I(a,b) + \frac{0.5^{a+b}}{aB(a,b)}, \tag{35}$$
$$I(a,b+1) = I(a,b) - \frac{0.5^{a+b}}{bB(a,b)}. \tag{36}$$

   The properties for $I(a,b)$ are derived from the basic property of regularized incomplete beta function. [1]

Proof [Proof of Lemma 10]

When $a = b$, by Corollary 3, we have $I(a+1,b) > 0.5$, $I(a,b) = 0.5$ and $I(a,b+1) < 0.5$. Therefore, the expected reward (30) takes the following form:

$$\begin{aligned} R(a,b) &= 0.5(I(a+1,a) - I(a,a)) + 0.5((1 - I(a,a+1)) - I(a,a)) \\ &= I(a+1,a) - I(a,a) = \frac{0.5^{2a}}{aB(a,a)}. \end{aligned}$$

1. http://dlmf.nist.gov/8.17

When $a > b$, since a, b are integers, we have $a \geq b+1$ and hence $I(a+1, b) > 0.5, I(a, b) > 0.5, I(a, b+1) \geq 0.5$ according to Corollary 3. The expected reward (30) now becomes:

$$\begin{aligned}
R(a,b) =& \frac{a}{a+b} I(a+1,b) + \frac{b}{a+b} I(a,b+1) - I(a,b) \\
=& \frac{a}{a+b} \frac{1}{B(a+1,b)} \int_{0.5}^{1} t \cdot t^{a-1}(1-t)^{b-1} \mathrm{d}t \\
&+ \frac{b}{a+b} \frac{1}{B(a,b+1)} \int_{0.5}^{1} t^{a-1}(1-t)(1-t)^{b-1} \mathrm{d}t - I(a,b) \\
=& \frac{1}{B(a,b)} \int_{0.5}^{1} (t + (1-t)) \cdot t^{a-1}(1-t)^{b-1} \mathrm{d}t - I(a,b) \\
=& I(a,b) - I(a,b) = 0.
\end{aligned}$$

Here we use (32) and (33) to show that $\frac{a}{a+b}\frac{1}{B(a+1,b)} = \frac{b}{a+b}\frac{1}{B(a,b+1)} = \frac{1}{B(a,b)}$.
When $a \leq b-1$, we can prove $R(a,b) = 0$ in a similar way. ■

With Lemma 10 in place, the proof for Proposition 6 is straightforward. Recall that the deterministic KG policy chooses the next instance according to

$$i_t = \arg\max_i R(S^t, i) = \arg\max_i R(a_i^t, b_i^t),$$

and breaks the tie by selecting the one with the smallest index. Since $R(a,b) > 0$ if and only if $a = b$, at the initial stage $t = 0$, $R(a_i^0, b_i^0) > 0$ for those instances $i \in \mathcal{E} = \{i : a_i^0 = b_i^0\}$. The policy will first select $i_0 \in \mathcal{E}$ with the largest $R(a_i^0, b_i^0)$. After obtaining the label y_{i_0}, either $a_{i_0}^0$ or $b_{i_0}^0$ will add one and hence $a_{i_0}^1 \neq b_{i_0}^1$ and $R(a_{i_0}^1, b_{i_0}^1) = 0$. The policy will select another instance $i_1 \in \mathcal{E}$ with the "current" largest expected reward and the expected reward for i_1 after obtaining the label y_{i_1} will then become zero. As a consequence, the KG policy will label each instance in $\mathcal{E}$ for the first $|\mathcal{E}|$ stages and $R(a_i^{|\mathcal{E}|}, b_i^{|\mathcal{E}|}) = 0$ for all $i \in \{1, \ldots, K\}$. Then the deterministic policy will break the tie selecting the first instance to label. From now on, for any $t \geq |\mathcal{E}|$, if $a_1^t \neq b_1^t$, then the expected reward $R(a_1^t, b_1^t) = 0$. Since the expected reward for other instances are all zero, the policy will still label the first instance. On the other hand, if $a_1^t = b_1^t$, and the first instance is the only one with the positive expected reward and the policy will label it. Thus Proposition 6 is proved.

Remark 11 *For randomized KG, after getting one label for each instance in $\mathcal{E}$ for the first $|\mathcal{E}|$ stages, the expected reward for each instance has become zero. Then randomized KG will uniformly select one instance to label. At any stage $t \geq |\mathcal{E}|$, if there exists one instance i (at most one instance) with $a_i^t = b_i^t$, the KG policy will provide the next label for i; otherwise, it will randomly select an instance to label.*

A.5 Proof of Theorem 7

To prove the consistency of the Opt-KG policy, we first show the exact values for $R_\alpha^+(a,b) = \max(R_1(a,b), R_2(a,b))$.

1. When $a \geq b+1$:

$$R_1(a,b) = I(a+1,b) - I(a,b) = \frac{0.5^{a+b}}{aB(a,b)} > 0;$$
$$R_2(a,b) = I(a,b+1) - I(a,b) = -\frac{0.5^{a+b}}{bB(a,b)} < 0.$$

   Therefore,

$$R^+(a,b) = R_1(a,b) = \frac{0.5^{a+b}}{aB(a,b)} > 0.$$

2. When $a = b$:

$$R_1(a,b) = I(a+1,a) - I(a,a) = \frac{0.5^{2a}}{aB(a,a)};$$
$$R_2(a,b) = 1 - I(a,a+1) - I(a,a) = \frac{0.5^{2a}}{aB(a,a)}.$$

   Therefore, we have $R_1 = R_2$ and

$$R^+(a,b) = R_1(a,b) = R_2(a,b) = \frac{0.5^{2a}}{aB(a,a)} > 0.$$

3. When $b-1 \geq a$:

$$R_1(a,b) = I(a,b) - I(a+1,b) = -\frac{0.5^{a+b}}{aB(a,b)} < 0;$$
$$R_2(a,b) = I(a,b) - I(a,b+1) = \frac{0.5^{a+b}}{bB(a,b)} > 0.$$

   Therefore

$$R^+(a,b) = R_2(a,b) = \frac{0.5^{a+b}}{bB(a,b)} > 0.$$

We note that the values of $R^+(a,b)$ for different a, b are plotted in Figure 2 in main text.

As we can see $R^+(a,b) > 0$, for any positive integers (a,b), we first prove in the following lemma that

$$\lim_{a+b\to\infty} R^+(a,b) = 0. \tag{37}$$

Lemma 12 *Properties for $R^+(a,b)$:*

1. *$R(a,b)$ is symmetric, i.e., $R^+(a,b) = R^+(b,a)$.*
2. *$\lim_{a\to\infty} R^+(a,a) = 0$.*
3. *For any fixed $a \geq 1$, $R^+(a+k, a-k) = R^+(a-k, a+k)$ is monotonically decreasing in k for $k = 0, \ldots, a-1$.*

4. *When $a \geq b$, for any fixed b, $R^+(a,b)$ is monotonically decreasing in a. By the symmetry of $R^+(a,b)$, when $b \geq a$, for any fixed a, $R^+(a,b)$ is monotonically decreasing in b.*

By the above four properties, we have $\lim_{(a+b)\to\infty} R^+(a,b) = 0$.

Proof [Proof of Lemma 12]
We first prove these four properties.

- Property 1: By the fact that $B(a,b) = B(b,a)$, the symmetry of $R^+(a,b)$ is straightforward.
- Property 2: For $a > 1$, $\frac{R^+(a,a)}{R^+(a-1,a-1)} = \frac{2a-1}{2a} < 1$ and hence $R^+(a,a)$ is monotonically decreasing in a. Moreover,

$$R^+(a,a) = R^+(1,1)\prod_{i=2}^{a}\frac{2i-1}{2i} = R^+(1,1)\prod_{i=2}^{a}(1-\frac{1}{2i}) \leq R^+(1,1)e^{-\sum_{i=2}^{a}\frac{1}{2i}}.$$

  Since $\lim_{a\to\infty}\sum_{i=2}^{a}\frac{1}{2i} = \infty$ and $R^+(a,a) \geq 0$, $\lim_{a\to\infty} R^+(a,a) = 0$.
- Property 3: For any $k \geq 0$,

$$\begin{aligned}\frac{R^+(a+(k+1),a-(k+1))}{R^+(a+k,a-k)} &= \frac{(a+k)B(a+k,a-k)}{(a+k+1)B(a+(k+1),a-(k+1))}\\ &= \frac{a-(k+1)}{a+(k+1)} < 1.\end{aligned}$$

- Property 4: When $a \geq b$, for any fixed b:

$$\frac{R^+(a+1,b)}{R^+(a,b)} = \frac{aB(a,b)}{2(a+1)B(a+1,b)} = \frac{a(a+b)}{2a(a+1)} < 1.$$

According to the third property, when $a+b$ is an even number, we have $R^+(a,b) < R^+(\frac{a+b}{2},\frac{a+b}{2})$. According to the fourth property, when $a+b$ is an odd number and $a \geq b+1$, we have $R^+(a,b) < R^+(a-1,b) < R^+(\frac{a+b-1}{2},\frac{a+b-1}{2})$; while when $a+b$ is an odd number and $a \leq b-1$, we have $R^+(a,b) < R^+(a,b-1) < R^+(\frac{a+b-1}{2},\frac{a+b-1}{2})$. Therefore,

$$R^+(a,b) < R^+\left(\lfloor\frac{a+b}{2}\rfloor,\lfloor\frac{a+b}{2}\rfloor\right).$$

According to the second property such that $\lim_{a\to\infty} R^+(a,a) = 0$, we obtain (37). ■

Using Lemma 12, we first show that, in any sample path, the Opt-KG will label each instance infinitely many times as T goes to infinity. Let $\eta_i(T)$ be a random variable representing the number of times that the i-th instance has been labeled until the stage T using Opt-KG. Given a sample path ω, let $\mathcal{I}(\omega) = \{i : \lim_{T\to\infty}\eta_i(T)(\omega) < \infty\}$ be the set of instances that has been labeled only finite number of times as T goes to infinity in

this sample path. We need to prove that $\mathcal{I}(\omega)$ is an empty set for any ω. We prove it by contradiction. Assuming that $\mathcal{I}(\omega)$ is not empty, then after a certain stage $\widehat{T}$, instances in $\mathcal{I}(\omega)$ will never be labeled. By Lemma 12, for any $j \in \mathcal{I}^c$, $\lim_{T\to\infty} R^+(a_j^T(\omega), b_j^T(\omega)) = 0$. Therefore, there will exist $\bar{T} > \widehat{T}$ such that:

$$\max_{j\in\mathcal{I}^c} R^+(a_j^{\bar{T}}(\omega), b_j^{\bar{T}}(\omega)) \quad < \quad \max_{i\in\mathcal{I}} R^+(a_i^{\widehat{T}}(\omega), b_i^{\widehat{T}}(\omega)) = \max_{i\in\mathcal{I}} R^+(a_i^{\bar{T}}(\omega), b_i^{\bar{T}}(\omega)).$$

Then according to the Opt-KG policy, the next instance to be labeled must be in $\mathcal{I}(\omega)$, which leads to the contradiction. Therefore, $\mathcal{I}(\omega)$ will be an empty set for any ω.

Let Y_i^s be the random variable which takes the value 1 if the s-th label of the i-th instance is 1 and the value -1 if the s-th label is 0. It is easy to see that $\mathbb{E}(Y_i^s|\theta_i) = \Pr(Y_i^s = 1|\theta_i) = \theta_i$. Hence, Y_i^s, $s = 1, 2, \ldots$ are independent and identically distributed random variables. By the fact that $\lim_{T\to\infty} \eta_T(i) = \infty$ in all sample paths and using the strong law of large number, we conclude that, conditioning on θ_i, $i = 1, \ldots, K$, the conditional probability of

$$\lim_{T\to\infty} \frac{a_i^T - b_i^T}{\eta_i(T)} = \lim_{T\to\infty} \frac{\sum_{s=1}^{\eta_i(T)} Y_i^s}{\eta_i(T)} = \mathbb{E}(Y_i^s|\theta_i) = 2\theta_i - 1$$

for all $i = 1, \ldots, K$, is one. According to Proposition 2, we have $H_T = \{i : a_i^T \geq b_i^T\}$ and $H^* = \{i : \theta_i \geq 0.5\}$. The accuracy is $\mathrm{Acc}(T) = \frac{1}{K}\left(|H_T \cap H^*| + |H_T^c \cap (H^*)^c|\right)$. We have:

$$\begin{aligned}
&\Pr(\lim_{T\to\infty} \mathrm{Acc}(T) = 1|\{\theta_i\}_{i=1}^K) = \Pr\left(\lim_{T\to\infty} (|H_T \cap H^*| + |H_T^c \cap (H^*)^c|) = K|\{\theta_i\}_{i=1}^K\right) \\
\geq &\Pr\left(\lim_{T\to\infty} \frac{a_i^T - b_i^T}{\eta_i(T)} = 2\theta_i - 1, \forall i = 1, \ldots, K|\{\theta_i\}_{i=1}^K\right) = 1,
\end{aligned}$$

whenever $\theta_i \neq 0.5$ for all i. The last inequality is due to the fact that, as long as θ_i is not 0.5 in any i, any sample path that gives the event $\lim_{T\to\infty} \frac{a_i^T - b_i^T}{\eta_i(T)} = 2\theta_i - 1, \forall i = 1, \ldots, K$ also gives the event $\lim_{T\to\infty}(a_i^T - b_i^T) = \mathrm{sgn}(2\theta_i - 1)(+\infty)$, which further implies $\lim_{T\to\infty}(|H_T \cap H^*| + |H_T^c \cap (H^*)^c|) = K$.

Finally, we have:

$$\begin{aligned}
\Pr\left(\lim_{T\to\infty} \mathrm{Acc}(T) = 1\right) &= \mathbb{E}_{\{\theta_i\}_{i=1}^K}\left[\Pr\left(\lim_{T\to\infty} \mathrm{Acc}(T) = 1|\{\theta_i\}_{i=1}^K\right)\right] \\
&= \mathbb{E}_{\{\theta_i:\theta_i\neq 0.5\}_{i=1}^K}\left[\Pr\left(\lim_{T\to\infty} \mathrm{Acc}(T) = 1|\{\theta_i\}_{i=1}^K\right)\right] \\
&= \mathbb{E}_{\{\theta_i:\theta_i\neq 0.5\}_{i=1}^K}[1] = 1,
\end{aligned}$$

where the second equality is because $\{\theta_i : \exists i, \theta_i = 0.5\}$ is a zero measure set.

A.6 Proof of Proposition 8

Recall that our random reward is a two-point distribution with the probability $p_1 = \frac{a}{a+b}$ of being $R_1(a, b) = h(I(a + 1, b)) - h(I(a, b))$ and $p_2 = \frac{b}{a+b}$ of being $R_2(a, b) = h(I(a, b + 1)) - h(I(a, b))$. The pessimistic KG selects the next instance which maximizes $R^-(a, b) =$

$\min(R_1(a,b), R_2(a,b))$. To show that the policy is inconsistent, we first compute the exact values for $R^-(a,b)$ for positive integers (a,b).

Utilizing Corollary 3 and the basic properties of $I(a,b)$ in (34), (35), (36), we have:

1. When $a \geq b+1$:

$$R_1(a,b) = I(a+1,b) - I(a,b) = \frac{0.5^{a+b}}{aB(a,b)} > 0;$$
$$R_2(a,b) = I(a,b+1) - I(a,b) = -\frac{0.5^{a+b}}{bB(a,b)} < 0.$$

   Therefore,

$$R^-(a,b) = R_2(a,b) = -\frac{0.5^{a+b}}{bB(a,b)} < 0.$$

2. When $a = b$:

$$R_1(a,b) = I(a+1,a) - I(a,a) = \frac{0.5^{2a}}{aB(a,a)};$$
$$R_2(a,b) = 1 - I(a,a+1) - I(a,a) = \frac{0.5^{2a}}{aB(a,a)}.$$

   Therefore, we have $x_1 = x_2$ and

$$R^-(a,b) = R_1(a,b) = R_2(a,b) = \frac{0.5^{2a}}{aB(a,a)} > 0.$$

3. When $b - 1 \geq a$:

$$R_1(a,b) = I(a,b) - I(a+1,b) = -\frac{0.5^{a+b}}{aB(a,b)} < 0;$$
$$R_2(a,b) = I(a,b) - I(a,b+1) = \frac{0.5^{a+b}}{bB(a,b)} > 0.$$

   Therefore

$$R^-(a,b) = R_1(a,b) = -\frac{0.5^{a+b}}{aB(a,b)} < 0.$$

We summarize the properties of $R^-(a,b)$ in the next Lemma.

Lemma 13 *Properties for $R^-(a,b)$:*

1. *$R^-(a,b) > 0$ if and only if $a = b$.*
2. *$R^-(a,b)$ is symmetric, i.e., $R^-(a,b) = R^-(b,a)$*
3. *When $a = b+1$, then $R^-(a,b) = R^-(b+1,b)$ is monotonically increasing in b. By the symmetry of $R^-(a,b)$, when $b = a+1$, $R^-(a,b) = R^-(a,a+1)$ is monotonically increasing in a.*

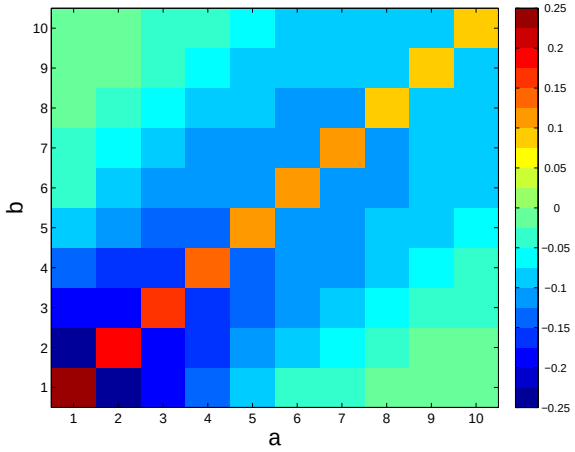


Figure 13: Illustration of $R^-(a,b)$.

4. *When $a \geq b+1$, for any fixed b, $R^-(a,b)$ is monotonically increasing in a. By the symmetry of $R^-(a,b)$, when $b \geq a+1$, for any fixed a, $R^-(a,b)$ is monotonically increasing in b.*

For better visualization, we plot values of $R^-(a,b)$ for different a,b in Figure 13. All the properties in Lemma 13 can be seen clearly from Figure 13. The proof of these properties are based on simple algebra and thus omitted here.

From Lemma 13, we can conclude that for any positive integers a,b with $a+b \neq 3$:

$$R^-(1,2) = R^-(2,1) < R^-(a,b). \tag{38}$$

Recall that the pessimistic KG selects:

$$i_t = \underset{i\in\{1,\ldots,K\}}{\arg\max}\ R^-(a_i^t, b_i^t).$$

When starting from the uniform prior with $a_i^0 = b_i^0 = 1$ for all $i \in \{1\ldots,K\}$, the corresponding $R^-(a_i^0, b_i^0) = R^-(1,1) > 0$. After obtaining a label for any instance i, the Beta parameters for θ_i will become either $(2,1)$ or $(1,2)$ with $R^-(1,2) = R^-(2,1) < 0$. Therefore, for the first K stages, the pessimistic KG policy will acquire the label for each instance once. For any instance i, we have either $a_i^K = 2, b_i^K = 1$ or $a_i^K = 1, b_i^K = 2$ at the stage K. Then the pessimistic KG policy will select the first instance to label. According to (38), for any $t \geq K$, $R^-(a_1^t, b_1^t) > R^-(1,2) = R^-(2,1)$. Therefore, the pessimistic KG policy will consistently acquire the label for the first instance. Since the tie will only appear at the stage K, the randomized pessimistic KG will also consistently select a single instance to label after K stages.

Appendix B. Incorporate Reliability of Heterogeneous Workers

As we discussed in Section 5 in main text, we approximate the posterior so that at any stage for all i,j, θ_i and ρ_j will follow Beta distributions. In particular, assuming at the

current state $\theta_i \sim \text{Beta}(a_i, b_i)$ and $\rho_j \sim \text{Beta}(c_j, d_j)$, the posterior distribution conditioned on z_{ij} takes the following form:

$$p(\theta_i, \rho_j | z_{ij} = 1) = \frac{\Pr(z_{ij} = 1|\theta_i, \rho_j)\text{Beta}(a_i, b_i)\text{Beta}(c_j, d_j)}{\Pr(z_{ij} = 1)},$$
$$p(\theta_i, \rho_j | z_{ij} = -1) = \frac{\Pr(z_{ij} = -1|\theta_i, \rho_j)\text{Beta}(a_i, b_i)\text{Beta}(c_j, d_j)}{\Pr(z_{ij} = -1)},$$

where the likelihood $\Pr(z_{ij} = z|\theta_i, \rho_j)$ for $z = 1, -1$ is defined in (18) and (19), i.e.,

$$\begin{aligned}\Pr(z_{ij} = 1|\theta_i, \rho_j) &= \theta_i\rho_j + (1 - \theta_i)(1 - \rho_j),\\ \Pr(z_{ij} = -1|\theta_i, \rho_j) &= (1 - \theta_i)\rho_j + \theta_i(1 - \rho_j).\end{aligned}$$

Also,

$$\begin{aligned}\Pr(z_{ij} = 1) &= \mathbb{E}(\Pr(z_{ij} = 1|\theta_i, \rho_j)) = \mathbb{E}(\theta_i)\mathbb{E}(\rho_j) + (1 - \mathbb{E}(\theta_i))(1 - \mathbb{E}(\rho_j))\\ &= \frac{a_i}{a_i + b_i}\frac{c_j}{c_j + d_j} + \frac{b_i}{a_i + b_i}\frac{d_j}{c_j + d_j},\end{aligned}$$

$$\begin{aligned}\Pr(z_{ij} = -1) &= \mathbb{E}(\Pr(z_{ij} = -1|\theta_i, \rho_j)) = (1 - \mathbb{E}(\theta_i))\mathbb{E}(\rho_j) + \mathbb{E}(\theta_i)(1 - \mathbb{E}(\rho_j))\\ &= \frac{b_i}{a_i + b_i}\frac{c_j}{c_j + d_j} + \frac{a_i}{a_i + b_i}\frac{d_j}{c_j + d_j}.\end{aligned}$$

The posterior distributions $p(\theta_i, p_j|z_{ij} = z)$ no longer takes the form of the product of Beta distributions on θ_i and p_j. Therefore, we use variational approximation by first assuming the conditional independence of θ_i and ρ_j:

$$p(\theta_i, \rho_j|z_{ij} = z) \approx p(\theta_i|z_{ij} = z)p(\rho_j|z_{ij} = z).$$

In fact, the exact form of marginal distributions can be calculated as follows:

$$p(\theta_i|z_{ij} = 1) = \frac{\theta_i\mathbb{E}(\rho_j) + (1 - \theta_i)(1 - \mathbb{E}(\rho_j))}{\Pr(z_{ij} = 1)}\text{Beta}(a_i, b_i),$$
$$p(\rho_j|z_{ij} = 1) = \frac{\mathbb{E}(\theta_i)\rho_j + (1 - \mathbb{E}(\theta_i))(1 - \rho_j)}{\Pr(z_{ij} = 1)}\text{Beta}(c_j, d_j),$$
$$p(\theta_i|z_{ij} = -1) = \frac{(1 - \theta_i)\mathbb{E}(\rho_j) + \theta_i(1 - \mathbb{E}(\rho_j))}{\Pr(z_{ij} = -1)}\text{Beta}(a_i, b_i),$$
$$p(\rho_j|z_{ij} = -1) = \frac{(1 - \mathbb{E}(\theta_i))\rho_j + \mathbb{E}(\theta_i)(1 - \rho_j)}{\Pr(z_{ij} = -1)}\text{Beta}(c_j, d_j).$$

To approximate the marginal distribution as Beta distribution, we use the moment matching technique. In particular, we approximate $p\left(\theta_i|z_{ij} = z\right) \approx \text{Beta}(\tilde{a}_i(z), \tilde{b}_i(z))$ such that

$$\widetilde{\mathbb{E}}_z(\theta_i) \doteq \mathbb{E}_{p(\theta_i|z_{ij}=z)}(\theta_i) = \frac{\tilde{a}_i(z)}{\tilde{a}_i(z) + \tilde{b}_i(z)}, \tag{39}$$
$$\widetilde{\mathbb{E}}_z(\theta_i^2) \doteq \mathbb{E}_{p(\theta_i|z_{ij}=z)}(\theta_i^2) = \frac{\tilde{a}_i(z)(\tilde{a}_i(z) + 1)}{(\tilde{a}_i(z) + \tilde{b}_i(z))(\tilde{a}_i(z) + \tilde{b}_i(z) + 1)}, \tag{40}$$

where $\frac{\tilde{a}_i(z)}{\tilde{a}_i(z)+\tilde{b}_i(z)}$ and $\frac{\tilde{a}_i(z)(\tilde{a}_i(z)+1)}{(\tilde{a}_i(z)+\tilde{b}_i(z))(\tilde{a}_i(z)+\tilde{b}_i(z)+1)}$ are the first and second order moment of $\text{Beta}(\tilde{a}_i(z), \tilde{b}_i(z))$. To make (39) and (40) hold, we have:

$$\tilde{a}_i(z) = \widetilde{\mathbb{E}}_z(\theta_i)\frac{\widetilde{\mathbb{E}}_z(\theta_i) - \widetilde{\mathbb{E}}_z(\theta_i^2)}{\widetilde{\mathbb{E}}_z(\theta_i^2) - \left(\widetilde{\mathbb{E}}_z(\theta_i)\right)^2}, \tag{41}$$

$$\tilde{b}_i(z) = (1 - \widetilde{\mathbb{E}}_z(\theta_i))\frac{\widetilde{\mathbb{E}}_z(\theta_i) - \widetilde{\mathbb{E}}_z(\theta_i^2)}{\widetilde{\mathbb{E}}_z(\theta_i^2) - \left(\widetilde{\mathbb{E}}_z(\theta_i)\right)^2}. \tag{42}$$

Similarly, we approximate $p\left(\rho_j | z_{ij} = z\right) \approx \text{Beta}(\tilde{c}_j(z), \tilde{d}_j(z))$, such that

$$\widetilde{\mathbb{E}}_z(\rho_j) \doteq \mathbb{E}_{p(\rho_j|z_{ij}=z)}(\rho_j) = \frac{\tilde{c}_j(z)}{\tilde{c}_j(z) + \tilde{d}_j(z)}, \tag{43}$$

$$\widetilde{\mathbb{E}}_z(\rho_j^2) \doteq \mathbb{E}_{p(\rho_j|z_{ij}=z)}(\rho_j^2) = \frac{\tilde{c}_j(z)(\tilde{c}_j(z)+1)}{(\tilde{c}_j(z) + \tilde{d}_j(z))(\tilde{c}_j(z) + \tilde{d}_j(z)+1)}, \tag{44}$$

where $\frac{\tilde{c}_j(z)}{\tilde{c}_j(z)+\tilde{d}_j(z)}$ and $\frac{\tilde{c}_j(z)(\tilde{c}_j(z)+1)}{(\tilde{c}_j(z)+\tilde{d}_j(z))(\tilde{c}_j(z)+\tilde{d}_j(z)+1)}$ are the first and second order moment of $\text{Beta}(\tilde{c}_j(z), \tilde{d}_j(z))$. To make (39) and (40) hold, we have:

$$\tilde{c}_j(z) = \widetilde{\mathbb{E}}_z(\rho_j)\frac{\widetilde{\mathbb{E}}_z(\rho_j) - \widetilde{\mathbb{E}}_z(\rho_j^2)}{\widetilde{\mathbb{E}}_z(\rho_j^2) - \left(\widetilde{\mathbb{E}}_z(\rho_j)\right)^2}, \tag{45}$$

$$\tilde{d}_j(z) = (1 - \widetilde{\mathbb{E}}_z(\rho_j))\frac{\widetilde{\mathbb{E}}_z(\rho_j) - \widetilde{\mathbb{E}}_z(\rho_j^2)}{\widetilde{\mathbb{E}}_z(\rho_j^2) - \left(\widetilde{\mathbb{E}}_z(\rho_j)\right)^2}. \tag{46}$$

Furthermore, we can compute the exact values for $\widetilde{\mathbb{E}}_z(\theta_i)$, $\widetilde{\mathbb{E}}_z(\theta_i^2)$, $\widetilde{\mathbb{E}}_z(\rho_j)$ and $\widetilde{\mathbb{E}}_z(\rho_j^2)$ as follows.

$$\widetilde{\mathbb{E}}_1(\theta_i) = \frac{\mathbb{E}(\theta_i^2)\mathbb{E}(\rho_j) + (\mathbb{E}(\theta_i) - \mathbb{E}(\theta_i^2))(1 - \mathbb{E}(\rho_j))}{p(z_{ij} = 1)} = \frac{a_i((a_i+1)c_j + b_i d_j)}{(a_i + b_i + 1)(a_i c_j + b_i d_j)},$$

$$\widetilde{\mathbb{E}}_1(\theta_i^2) = \frac{\mathbb{E}(\theta_i^3)\mathbb{E}(\rho_j) + (\mathbb{E}(\theta_i^2) - \mathbb{E}(\theta_i^3))(1 - \mathbb{E}(\rho_j))}{p(z_{ij} = 1)} = \frac{a_i(a_i+1)((a_i+2)c_j + b_i d_j)}{(a_i + b_i + 1)(a_i + b_i + 2)(a_i c_j + b_i d_j)},$$

$$\widetilde{\mathbb{E}}_{-1}(\theta_i) = \frac{(\mathbb{E}(\theta_i) - \mathbb{E}(\theta_i^2))\mathbb{E}(\rho_j) + \mathbb{E}(\theta_i^2)(1 - \mathbb{E}(\rho_j))}{p(z_{ij} = -1)} = \frac{a_i(b_i c_j + (a_i+1)d_j)}{(a_i + b_i + 1)(b_i c_j + a_i d_j)},$$

$$\widetilde{\mathbb{E}}_{-1}(\theta_i^2) = \frac{(\mathbb{E}(\theta_i^2) - \mathbb{E}(\theta_i^3))\mathbb{E}(\rho_j) + \mathbb{E}(\theta_i^3)(1 - \mathbb{E}(\rho_j))}{p(z_{ij} = -1)} = \frac{a_i(a_i+1)(b_i c_j + (a_i+2)d_j)}{(a_i + b_i + 1)(a_i + b_i + 2)(b_i c_j + a_i d_j)},$$

$$\widetilde{\mathbb{E}}_1(\rho_j) = \frac{\mathbb{E}(\theta_i)\mathbb{E}(\rho_j^2) + (1 - \mathbb{E}(\theta_i))(\mathbb{E}(\rho_j) - \mathbb{E}(\rho_j^2))}{p(z_{ij} = 1)} = \frac{c_j(a_i(c_j+1) + b_i d_j)}{(c_j + d_j + 1)(a_i c_j + b_i d_j)},$$

$$\widetilde{\mathbb{E}}_1(\rho_j^2) = \frac{\mathbb{E}(\theta_i)\mathbb{E}(\rho_j^3) + (1 - \mathbb{E}(\theta_i))(\mathbb{E}(\rho_j^2) - \mathbb{E}(\rho_j^3))}{p(z_{ij} = 1)} = \frac{c_j(c_j+1)(a_i(c_j+2) + b_i d_j)}{(c_j + d_j + 1)(c_j + d_j + 2)(a_i c_j + b_i d_j)},$$

$$\widetilde{\mathbb{E}}_{-1}(\rho_j) = \frac{(1 - \mathbb{E}(\theta_i))\mathbb{E}(\rho_j^2) + \mathbb{E}(\theta_i)(\mathbb{E}(\rho_j) - \mathbb{E}(\rho_j^2))}{p(z_{ij} = -1)} = \frac{c_j(b_i(c_j+1) + a_i d_j)}{(c_j + d_j + 1)(b_i c_j + a_i d_j)},$$

$$\widetilde{\mathbb{E}}_{-1}(\rho_j^2) = \frac{(1 - \mathbb{E}(\theta_i))\mathbb{E}(\rho_j^3) + \mathbb{E}(\theta_i)(\mathbb{E}(\rho_j^2) - \mathbb{E}(\rho_j^3))}{p(z_{ij} = -1)} = \frac{c_j(c_j+1)(b_i(c_j+2) + a_i d_j)}{(c_j + d_j + 1)(c_j + d_j + 2)(b_i c_j + a_i d_j)}.$$

Assuming that at a certain stage, θ_i follows a Beta posterior $\text{Beta}(a_i, b_i)$ and ρ_j follows a Beta posterior $\text{Beta}(c_j, d_j)$, the reward of getting positive and negative labels for the i-th instance from the j-th worker are:

$$R_1(a_i, b_i, c_j, d_j) = h(I(\tilde{a}_i(z=1), \tilde{b}_i(z=1))) - h(I(a_i, b_i)), \tag{47}$$

$$R_2(a_i, b_i, c_j, d_j) = h(I(\tilde{a}_i(z=-1), \tilde{b}_i(z=-1))) - h(I(a_i, b_i)), \tag{48}$$

where $\tilde{a}_i(z=\pm 1)$ and $\tilde{b}_i(z=\pm 1)$ are defined in (41) and (42), which further depend on c_j and d_j through $\widetilde{\mathbb{E}}_z(\theta_i)$ and $\widetilde{\mathbb{E}}_z(\theta_i^2)$. With the reward in place, we can directly apply the Opt-KG policy in the heterogeneous worker setting.

Appendix C. Extensions

In this section, we provide detailed Opt-KG algorithms for extensions in Section 6.

C.1 Utilizing Contextual Information

When each instance is associated with a p-dimensional feature vector $\mathbf{x}_i \in \mathbb{R}^p$, we incorporate the feature information in our budget allocation problem by assuming:

$$\theta_i = \sigma(\langle \mathbf{w}, \mathbf{x}_i \rangle) \doteq \frac{1}{1 + \exp\{-\langle \mathbf{w}, \mathbf{x}_i \rangle\}}, \tag{49}$$

where $\sigma(x) = \frac{1}{1+\exp\{-x\}}$ is the sigmoid function and $\mathbf{w}$ is assumed to be drawn from a Gaussian prior $N(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$. At the t-th stage with the state $S^t = (\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$ and $\mathbf{w} \sim (\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$, the decision maker chooses the i_t-th instance to be labeled and observes the label $y_{i_t} \in \{-1, 1\}$. The posterior distribution $p(\mathbf{w}|y_{i_t}, S^t) \propto p(y_{i_t}|\mathbf{w})p(\mathbf{w}|S^t)$ has the following log-likelihood:

$$\begin{aligned}
&\ln p(\mathbf{w}|y_{i_t}, S^t) = \ln p(y_{i_t}|\mathbf{w}) + \ln p(\mathbf{w}|S^t) + \text{const} \\
=&\mathbf{1}(y_{i_t} = 1) \ln \sigma(\langle \mathbf{w}, \mathbf{x}_{i_t} \rangle) + \mathbf{1}(y_{i_t} = -1) \ln (1 - \sigma(\langle \mathbf{w}, \mathbf{x}_{i_t} \rangle)) \\
&- \frac{1}{2}(\mathbf{w} - \boldsymbol{\mu}_t)' \boldsymbol{\Omega}_t (\mathbf{w} - \boldsymbol{\mu}_t) + \text{const},
\end{aligned}$$

where $\boldsymbol{\Omega}_t = (\boldsymbol{\Sigma}_t)^{-1}$ is the precision matrix. To approximate $p(\mathbf{w}|y_{i_t}, \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$ by a Gaussian distribution $N(\boldsymbol{\mu}_{t+1}, \boldsymbol{\Sigma}_{t+1})$, we use the Laplace method (see Chapter 4.4 in Bishop, 2007). In particular, the mean of the posterior Gaussian is the MAP (maximum a posteriori) estimator of $\mathbf{w}$:

$$\mu_{t+1} = \arg\max_{\mathbf{w}} \ln p(\mathbf{w}|y_{i_t}, S^t), \tag{50}$$

which can be computed by any numerical optimization method (e.g., Newton's method). The precision matrix takes the following form,

$$\boldsymbol{\Omega}_{t+1} = -\nabla^2 \ln p(\mathbf{w}|y_{i_t}, S^t)\big|_{\mathbf{w}=\boldsymbol{\mu}_{t+1}} = \boldsymbol{\Omega}_t + \sigma(\boldsymbol{\mu}'_{t+1}\mathbf{x}_{i_{t+1}})(1 - \sigma(\boldsymbol{\mu}'_{t+1}\mathbf{x}_{i_{t+1}}))\mathbf{x}_{i_{t+1}}\mathbf{x}'_{i_{t+1}}.$$

By Sherman-Morrison formula, the covariance matrix can be computed as,

$$\boldsymbol{\Sigma}_{t+1} = (\boldsymbol{\Omega}_{t+1})^{-1} = \boldsymbol{\Sigma}_t - \frac{\sigma(\boldsymbol{\mu}'_{t+1}\mathbf{x}_{i_t})(1 - \sigma(\boldsymbol{\mu}_{t+1}\mathbf{x}_{i_t}))}{1 + \sigma(\boldsymbol{\mu}'_{t+1}\mathbf{x}_{i_t})(1 - \sigma(\boldsymbol{\mu}'_{t+1}\mathbf{x}_{i_t}))\mathbf{x}'_{i_t}\boldsymbol{\Sigma}_t\mathbf{x}_{i_t}} \boldsymbol{\Sigma}_t \mathbf{x}_{i_{t+1}} \mathbf{x}'_{i_t} \boldsymbol{\Sigma}_t.$$

We also calculate the transition probability of $y_{i_t} = 1$ and $y_{i_t} = -1$ using the technique from Bayesian logistic regression (see Chapter 4.5 in Bishop, 2007):

$$\Pr(y_{i_t} = 1|S^t, i_t) = \int p(y_{i_t} = 1|\mathbf{w})p(\mathbf{w}|S^t)\mathrm{d}\mathbf{w} = \int \sigma(\mathbf{w}'\mathbf{x}_i)p(\mathbf{w}|S^t)\mathrm{d}\mathbf{w} \approx \sigma(\mu_i\kappa(s_i^2)),$$

where $\kappa(s_i^2) = (1 + \pi s_i^2/8)^{-1/2}$ and $\mu_i = \langle \boldsymbol{\mu}_t, \mathbf{x}_i \rangle$ and $s_i^2 = \mathbf{x}_i'\boldsymbol{\Sigma}_t\mathbf{x}_i$.

To calculate the reward function, in addition to the transition probability, we also need to compute:

$$\begin{aligned}
P_i^t &= \Pr(\theta_i \geq 0.5|\mathcal{F}_t) \\
&= \Pr\left(\frac{1}{1+\exp\{-\mathbf{w}_t'\mathbf{x}_i\}} \geq 0.5 \Big| \mathbf{w}_t \sim N(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)\right) \\
&= \Pr(\mathbf{w}_t'\mathbf{x}_i \geq 0|\mathbf{w}_t \sim N(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)) \\
&= \int_0^\infty \left(\int_{\mathbf{w}} \delta(c - \langle \mathbf{w}, \mathbf{x}_i \rangle) N(\mathbf{w}|\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)\mathrm{d}\mathbf{w}\right)\mathrm{d}c,
\end{aligned}$$

where $\delta(\cdot)$ is the Dirac delta function. Let

$$p(c) = \int_{\mathbf{w}} \delta(c - \langle \mathbf{w}, \mathbf{x}_i \rangle) N(\mathbf{w}|\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)\mathrm{d}\mathbf{w}.$$

Since the marginal of a Gaussian distribution is still a Gaussian, $p(c)$ is a univariate-Gaussian distribution with the mean and variance:

$$\begin{aligned}
\mu_i &= \mathbb{E}(c) = \langle \mathbb{E}(\mathbf{w}), \mathbf{x}_i \rangle = \langle \boldsymbol{\mu}_t, \mathbf{x}_i \rangle, \\
s_i^2 &= \mathrm{Var}(c) = (\mathbf{x}_i)'\mathrm{Cov}(\mathbf{w}, \mathbf{w})\mathbf{x}_i = (\mathbf{x}_i)'\boldsymbol{\Sigma}_t\mathbf{x}_i.
\end{aligned}$$

Therefore, we have:

$$P_i^t = \int_0^\infty p(c)\mathrm{d}c = 1 - \Phi\left(-\frac{\mu_i}{s_i}\right), \tag{51}$$

where $\Phi(\cdot)$ is the CDF of the standard Gaussian distribution.

With P_i^t and transition probability in place, the expected reward in value function takes the following form :

$$R(S^t, i_t) = \mathbb{E}\left(\sum_{i=1}^K h(P_i^{t+1}) - \sum_{i=1}^K h(P_i^t) \Big| S^t, i_t\right). \tag{52}$$

We note that since $\mathbf{w}$ will affect all P_i^t, the summation from 1 to K in (52) can not be omitted and hence (52) cannot be written as $\mathbb{E}\left(h(P_{i_t}^{t+1}) - h(P_{i_t}^t)|S^t, i_t\right)$ in (28). In this problem, KG or Opt-KG need to solve $O(2TK)$ optimization problems to compute the mean of the posterior as in (50), which could be computationally quite expensive. One possibility to address this problem is to use the variational Bayesian logistic regression (Jaakkola and Jordan, 2000), which could lead to a faster optimization procedure.

C.2 Multi-Class Categorization

Given the model and notations introduced in Section 6.2, at the final stage T when all budget is used up, we construct the set H_c^T for each class c to maximize the conditional expected classification accuracy:

$$\begin{aligned}\{H_c^T\}_{c=1}^C &= \underset{H_c\subseteq\{1,\ldots,C\},H_c\cap H_{\tilde{c}}=\emptyset}{\arg\max}\ \mathbb{E}\left(\sum_{i=1}^K\sum_{c=1}^C I(i\in H_c)I(i\in H_c^*)\Big|\mathcal{F}_T\right)\\ &= \underset{H_c\subseteq\{1,\ldots,C\},H_c\cap H_{\tilde{c}}=\emptyset}{\arg\max}\ \sum_{i=1}^K\sum_{c=1}^C I(i\in H_c)\Pr\left(i\in H_c^*|\mathcal{F}_T\right). \end{aligned} \tag{53}$$

Here, $H_c^* = \{i : \theta_{ic} \geq \theta_{ic'}, \forall c' \neq c\}$ is the true set of instances in the class c. The set H_c^T consists of instances that belong to class c. Therefore, $\{H_c^T\}_{c=1}^C$ should form a partition of all instances $\{1,\ldots,K\}$. Let

$$P_{ic}^T = \Pr(i\in H_c^*|\mathcal{F}_T) = \Pr(\theta_{ic}\geq\theta_{i\tilde{c}},\ \ \forall\ \ \tilde{c}\neq c|\mathcal{F}_T). \tag{54}$$

To maximize the right hand side of (53), we have

$$H_c^T = \{i : P_{ic}^T \geq P_{i\tilde{c}}^T, \forall\tilde{c}\neq c\}. \tag{55}$$

If there is i belongs to more than one H_c^T, we only assign it to the one with the smallest index c. The maximum conditional expected accuracy takes the form: $\sum_{i=1}^K\left(\max_{c\in\{1\ldots,C\}} P_{ic}^T\right)$.

Then the value function can be defined as:

$$V(S^0) \doteq \sup_\pi \mathbb{E}^\pi\left(\mathbb{E}\left(\sum_{i=1}^K\sum_{c=1}^C I(i\in H_c^T)I(i\in H_c^*)\Big|\mathcal{F}_T\right)\right) = \sup_\pi \mathbb{E}^\pi\left(\sum_{i=1}^K h(\mathbf{P}_i^T)\right),$$

where $\mathbf{P}_i^T = (P_{i1}^T,\ldots,P_{iC}^T)$ and $h(\mathbf{P}_i^T) \doteq \max_{c\in\{1\ldots,C\}} P_{ic}^T$. Following Proposition 4, let $P_{ic}^t = \Pr(i\in H_c^*|\mathcal{F}_t)$ and $\mathbf{P}_i^t = (P_{i1}^t,\ldots,P_{iC}^t)$, we define incremental reward function at each stage:

$$R(S^t, i_t) = \mathbb{E}\left(h(\mathbf{P}_{i_t}^{t+1}) - h(\mathbf{P}_{i_t}^t)|S^t, i_t\right).$$

The value function can be re-written as:

$$V(S^0) = G_0(S^0) + \sup_\pi \mathbb{E}^\pi\left(\sum_{t=0}^{T-1} R(S^t,i_t)\Big|S^0\right),$$

where $G_0(S^0) = \sum_{i=1}^K h(\mathbf{P}_i^0)$. Since the reward function only depends on $S_{i_t}^t = \boldsymbol{\alpha}_{i_t}^t \in \mathbb{R}_+^C$, we can define the reward function in a more explicit way by defining:

$$R(\boldsymbol{\alpha}) = \sum_{c=1}^C \frac{\alpha_c}{\sum_{\tilde{c}=1}^C \alpha_{\tilde{c}}} h(I(\boldsymbol{\alpha}+\boldsymbol{\delta}_c)) - h(I(\boldsymbol{\alpha})).$$

Here $\boldsymbol{\delta}_c$ be a row vector of length C with one at the c-th entry and zeros at all other entries; and $I(\boldsymbol{\alpha}) = (I_1(\boldsymbol{\alpha}), \ldots, I_C(\boldsymbol{\alpha}))$ where

$$I_c(\boldsymbol{\alpha}) = \Pr(\theta_c \geq \theta_{\tilde{c}}, \forall \tilde{c} \neq c | \theta \sim \mathrm{Dir}(\boldsymbol{\alpha})). \tag{56}$$

Therefore, we have $R(S^t, i_t) = R(\boldsymbol{\alpha}_{i_t}^t)$.

To evaluate the reward $R(\boldsymbol{\alpha})$, the major bottleneck is how to compute $I_c(\boldsymbol{\alpha})$ efficiently. Directly taking the C-dimensional integration on the region $\{\theta_c \geq \theta_{\tilde{c}}, \forall \tilde{c} \neq c\} \cap \Delta_C$ will be computationally very expensive, where Δ_C denotes the C-dimensional simplex. Therefore, we propose a method to convert the computation of $I_c(\boldsymbol{\alpha})$ into a one-dimensional integration. It is known that to generate $\theta \sim \mathrm{Dir}(\boldsymbol{\alpha})$, it is equivalent to generate $\{X_c\}_{c=1}^C$ with $X_c \sim \mathrm{Gamma}(\alpha_c, 1)$ and let $\theta_c \equiv \frac{X_c}{\sum_{c=1}^C X_c}$. Then $\theta = (\theta_1, \ldots, \theta_C)$ will follow $\mathrm{Dir}(\boldsymbol{\alpha})$. Therefore, we have:

$$I_c(\boldsymbol{\alpha}) = \Pr(X_c \geq X_{\tilde{c}}, \forall \tilde{c} \neq c | X_c \sim \mathrm{Gamma}(\alpha_c, 1)). \tag{57}$$

It is easy to see that

$$\begin{aligned} I_c(\boldsymbol{\alpha}) &= \int_{0 \leq x_1 \leq x_c} \cdots \int_{x_c \geq 0} \cdots \int_{0 \leq x_C \leq x_c} \prod_{c=1}^{C} f_{\mathrm{Gamma}}(x_c; \alpha_c, 1) \mathrm{d}x_1 \ldots \mathrm{d}x_C \\ &= \int_{x_c \geq 0} f_{\mathrm{Gamma}}(x_c; \alpha_c, 1) \prod_{\tilde{c} \neq c} F_{\mathrm{Gamma}}(x_c; \alpha_{\tilde{c}}, 1) \mathrm{d}x_c, \end{aligned} \tag{58}$$

where $f_{\mathrm{Gamma}}(x; \alpha_c, 1)$ is the density function of Gamma distribution with the parameter $(\alpha_c, 1)$ and $F_{\mathrm{Gamma}}(x_c; \alpha_{\tilde{c}}, 1)$ is the CDF of Gamma distribution at x_c with the parameter $(\alpha_{\tilde{c}}, 1)$. In many softwares, $F_{\mathrm{Gamma}}(x_c; \alpha_{\tilde{c}}, 1)$ can be calculated very efficiently without an explicit integration. Therefore, we can evaluate $I_c(\boldsymbol{\alpha})$ by performing only a one-dimensional numerical integration as in (58). We could also use Monte-Carlo approximation to further accelerate the computation in (58).

References

P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47:235–256, 2002.

Y. Bachrach, T. Minka, J. Guiver, and T. Graepel. How to grade a test without knowing the answers - a Bayesian graphical model for adaptive crowdsourcing and aptitude testing. In *ICML*, 2012.

M. J. Beal. *Variational Algorithms for Approximate Bayesian Inference*. PhD thesis, Gatsby Computational Neuroscience Unit, University College London, 2003.

C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2007.

S. Bubeck and N. Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012.

A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Journal of the Royal Statistical Society Series C*, 28:20–28, 1979.

S. Ertekin, H. Hirsh, and C. Rudin. Wisely using a budget for crowdsourcing. Technical report, MIT, 2012.

Eyal Even-Dar and Yishay Mansour. Convergence of optimistic and incremental Q-learning. In *NIPS*, 2001.

P. Frazier, W. B. Powell, and S. Dayanik. A knowledge-gradient policy for sequential information collection. *SIAM J. Control Optim.*, 47(5):2410–2439, 2008.

C. Gao and D. Zhou. Minimax optimal convergence rates for estimating ground truth from crowdsourced labels. arXiv:1310.5764, 2013.

Andrew Gelman, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, and Donald B. Rubin. *Bayesian Data Analysis*. Chapman and Hall, 3rd edition, 2013.

J. C. Gittins. *Multi-armed Bandit Allocation Indices.* John Wiley & Sons, 1989.

S. S. Gupta and K. J. Miescke. Bayesian look ahead one stage sampling allocations for selection the largest normal mean. *J. of Stat. Planning and Inference*, 54(2):229–244, 1996.

C. Ho, S. Jabbari, and J. W. Vaughan. Adaptive task assignment for crowdsourced classification. In *ICML*, 2013.

P. G. Ipeirotis, F. Provost, V. S. Sheng, and J. Wang. Repeated labeling using multiple noisy label. *Data Mining and Knowledge Discovery*, 2013.

T. Jaakkola and M. I. Jordan. Bayesian parameter estimation via variational methods. *Statistics and Computing*, 10:25–37, 2000.

E. Kamar, S. Hacker, and E. Horvitz. Combing human and machine intelligence in large-scale crowdsourcing. In *AAMAS*, 2012.

D. Karger, S. Oh, and D. Shah. Efficient crowdsourcing for multi-class labeling. In *ACM Sigmetrics*, 2013a.

D. R. Karger, S. Oh, and D. Shah. Budget-optimal task allocation for reliable crowdsourcing systems. *Operations Research*, 62(1):1–24, 2013b.

E. Kaufmann, O. Cappe, and A. Garivier. On Bayesian upper confidence bounds for bandit problems. In *AISTATS*, 2012.

L. Li, W. Chu, J. Langford, and R. E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of International World Wide Web Conference*, 2010.

C. Liu and Y. M. Wang. Truelabel + confusions: A spectrum of probabilistic models in analyzing multiple ratings. In *ICML*, 2012.

Q. Liu, J. Peng, and A. Ihler. Variational inference for crowdsourcing. In *NIPS*, 2012.

Andrew Y. Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *ICML*, 1999.

J. Nino-Mora. Computing a classic index for finite-horizon bandits. *INFORMS Journal on Computing*, 23(2):254–267, 2011.

R. D. Nowak. Noisy generalized binary search. In *NIPS*, 2009.

J. Paisley, D. Blei, and M. Jordan. Variational bayesian inference with stochastic search. In *ICML*, 2012.

W. B. Powell. *Approximate Dynamic Programming: solving the curses of dimensionality.* John Wiley & Sons, 2007.

M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* Wiley, 2005.

V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy. Learning from crowds. *Journal of Machine Learning Research*, 11:1297–1322, 2010.

C. P. Robert. *The Bayesian Choice: From Decision-Theoretic Foundations to Computational Implementation.* Springer, 2007.

R. T. Rockafellar and S. Uryasev. Conditional value-at-risk for general loss distributions. *J. of Banking and Finance*, 26:1443–1471, 2002.

B. Settles. Active learning literature survey. Technical report, University of Wisconsin–Madison, 2009.

R. Snow, B. O. Connor, D. Jurafsky, and A. Y. Ng. Cheap and fast - but is it good? evaluating non-expert annotations for natural language tasks. In *EMNLP*, 2008.

István Szita and András Lőrincz. The many faces of optimism: a unifying approach. In *ICML*, 2008.

J. Wang, P. G. Ipeirotis, and F. Provost. Quality-based pricing for crowdsourced workers. Technical report, New York University, 2013.

P. Welinder, S. Branson, S. Belongie, and P. Perona. The multidimensional wisdom of crowds. In *NIPS*, 2010.

J. Whitehill, P. Ruvolo, T. Wu, J. Bergsma, and J. R. Movellan. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *NIPS*, 2009.

J. Xie and P. I. Frazier. Sequential bayes-optimal policies for multiple comparisons with a known standard. *Operations Research*, 61(5):1174–1189, 2013.

Y. Yan, R. Rosales, G. Fung, and J. Dy. Active learning from crowds. In *ICML*, 2011.

D. Zhou, S. Basu, Y. Mao, and J. Platt. Learning from the wisdom of crowds by minimax conditional entropy. In *NIPS*, 2012.

Journal of Machine Learning Research 16 (2015) 47-75 Submitted 2/13; Revised 7/14; Published 1/15

Simultaneous Pursuit of Sparseness and Rank Structures for Matrix Decomposition

Qi Yan YANXX195@UMN.EDU
School of Statistics
University of Minnesota
Minneapolis, MN 55414, USA

Jieping Ye JIEPING.YE@ASU.EDU
Computer Science and Engineering
Arizona State University
Tempe, AZ 85287 USA

Xiaotong Shen XSHEN@UMN.EDU
School of Statistics
University of Minnesota
Minneapolis, MN 55414, USA

Editor: Aapo Hyvarinen

Abstract

In multi-response regression, pursuit of two different types of structures is essential to battle the curse of dimensionality. In this paper, we seek a sparsest decomposition representation of a parameter matrix in terms of a sum of sparse and low rank matrices, among many overcomplete decompositions. On this basis, we propose a constrained method subject to two nonconvex constraints, respectively for sparseness and low- rank properties. Computationally, obtaining an exact global optimizer is rather challenging. To overcome the difficulty, we use an alternating directions method solving a low-rank subproblem and a sparseness subproblem alternatively, where we derive an exact solution to the low-rank subproblem, as well as an exact solution in a special case and an approximated solution generally through a surrogate of the L_0-constraint and difference convex programming, for the sparse subproblem. Theoretically, we establish convergence rates of a global minimizer in the Hellinger-distance, providing an insight into why pursuit of two different types of decomposed structures is expected to deliver higher estimation accuracy than its counterparts based on either sparseness alone or low-rank approximation alone. Numerical examples are given to illustrate these aspects, in addition to an application to facial imagine recognition and multiple time series analysis.

Keywords: blockwise decent, nonconvex minimization, matrix decomposition, structure pursuit

1. Introduction

In multivariate analysis, data as well as parameters are usually expressed in terms of a matrix form, as opposed to a vector representation in univariate analysis. This occurs frequently in multi-class classification (Amit et al., 2007), matrix completion (Cai et al., 2010; Jain et al., 2010), collaborative filtering (Srebro et al., 2005), computer vision (Wright,

2009), among others. In situations as such, it essential to identify and employ certain lower-dimensional structures to battle the curse of dimensionality due to an increase in dimensionality from multivariate attributes. In this article, we explore rank and sparseness structures through matrix decomposition simultaneously in estimating large matrices through a novel notation of seeking a sparsest decomposition from a class of overcomplete decompositions.

Statistically, different structures have dramatically different interpretations. A low rank property of a matrix describes global information across different tasks, whereas sparseness concerns local information of specific task. For instance, for face images, the global information corresponds to the overall shape of a face, but the local information characterizes specific facial expression such as laugh and cry. In linear time-invariant (LTI) system, a low rank property corresponds to a low-order LTI system and a sparseness property captures an LTI system with a sparse impulse response (Porat, 1997). In a high-dimensional situation, betting on one type of structure may not be adequate to battle the curse of dimensionality. In this article, we seek a sparsest decomposition for the purpose of dimension reduction, from a class of overcomplete decompositions into simpler sparse and low-rank components. Specifically, a matrix $\boldsymbol{\Theta}$ is decomposed as $\boldsymbol{\Theta}_1 + \boldsymbol{\Theta}_2$, for a sparse $\boldsymbol{\Theta}_1$ and low-rank $\boldsymbol{\Theta}_2$ components, where $\boldsymbol{\Theta}_1$and $\boldsymbol{\Theta}_2$ are chosen from many such decompositions, with a smallest effective degrees of freedom, leading to high accuracy of parameter estimation. Our objective is to reconstruct the parameter matrix by identifying a sparsest decomposition consisting of simpler components. Such a decomposition can be used to provide a simpler and more efficient description of a complex system in terms of its simpler components. This results in more efficient structure representations leading to higher accuracy of parameter estimation in high-dimensional data analysis.

In this paper, we consider a multi-response linear regression problem in which a random sample $(\boldsymbol{a}_i, \boldsymbol{z}_i)_{i=1}^n$ is observed with a k-dimensional response vector $\boldsymbol{z}_i$ following

$$\boldsymbol{z}_i = \boldsymbol{a}_i^T \boldsymbol{\Theta} + \boldsymbol{\epsilon}_i, \quad E\boldsymbol{\epsilon}_i = 0,\ Cov(\boldsymbol{\epsilon}_i) = \sigma^2 \boldsymbol{I}; \quad i = 1, \ldots, n, \tag{1}$$

where $\boldsymbol{a}_i$ is a p-dimensional design vector, is independent of random error ϵ_i, and $\boldsymbol{I}$ is the identity matrix. Model (1) reduces to the univariate case when $k = 1$, and becomes a multivariate autoregressive model when $\boldsymbol{a}_i = \boldsymbol{z}_{i-1}$. Through matrix decomposition, we decompose a $p \times k$ regression parameter matrix $\boldsymbol{\Theta}$ into a sum of a sparse matrix $\boldsymbol{\Theta}_1$ and a low rank matrix $\boldsymbol{\Theta}_2$ for structure exploration, that is, $\boldsymbol{\Theta} = \boldsymbol{\Theta}_1 + \boldsymbol{\Theta}_2$. Model (1) is expressible in a matrix form

$$\boldsymbol{Z} = \boldsymbol{A}\boldsymbol{\Theta} + \boldsymbol{e}; \tag{2}$$

where $\boldsymbol{Z} = (\boldsymbol{z}_1, \cdots, \boldsymbol{z}_n)^T \in \mathbb{R}^{n \times k}$, $\boldsymbol{A} = (a_1, \cdots, a_n)^T$ is a $n \times p$ matrix, and $\boldsymbol{e} = (\boldsymbol{\epsilon_1}, \boldsymbol{\cdots}, \boldsymbol{\epsilon_n})^T \in \mathbb{R}^{n \times k}$ are the data, design and error matrices. In (1), we estimate $\boldsymbol{\Theta}$ based on n paired observation vectors $(\boldsymbol{a}_i, \boldsymbol{z}_i)_{i=1}^n$, with prior knowledge that $\boldsymbol{\Theta}_1$ is sparse in the number of its nonzero entries, and rank $r(\boldsymbol{\Theta}_2)$ is low relative to $\min(n, k, p)$. Our goal is to recover the parameter $\boldsymbol{\Theta}$ by identifying $\boldsymbol{\Theta}_1$ and $\boldsymbol{\Theta}_2$.

In the literature, the simultaneous exploration of rank and sparseness structures through matrix decomposition has received some attention, yet has not been well-studied. For robust principal component analysis (RPCA) where $\boldsymbol{A} = \boldsymbol{I}_{n \times p}$ is the $n \times p$ identity matrix with its

diagonals and off-diagonals being one and zero, Yuan & Yang (2013) and Chandrasekaran et al. (2011) employed a linear combination of the L_1 sparsity regularization and the nuclear-norm regularization, and Zhou & Tao (2011) used a randomized projections based low rank approximations and thresholding for sparsity pursuit. Moreover, Wright et al. (2013) recovers the sparse and low-rank components by minimizing a linear combination of the L_1-norm for sparsity and the nuclear-norm for low rank pursuit, while Waters et al. (2011) develops a greedy algorithm to pursue the sparse and low rank structures. For multiple task learning, Chen et al. (2010) studies sparse and low rank structures separately through convex regularization. In essence, most the existing literature focuses exclusively on a unique matrix decomposition of $\boldsymbol{\Theta}$ with $\boldsymbol{A} = \boldsymbol{I}_{n\times p}$ or $\boldsymbol{A}$ to be a set of random linear measurements, and without noise or with small noise that is essentially ignorable. For instance, Chandrasekaran et al. (2011) provided sufficient conditions for exact recovery of a convex relaxation method without noise; Wright et al. (2013) proved that recovering a target matrix is possible from a small set of randomly selected linear measurements when the number of measurements is sufficiently large. Among these, Agarwal et al. (2012) considered a general $\boldsymbol{A}$ and derived a theorem that bounds the Frobenius-norm error obtained through regularized convex relaxation under a "spikiness" condition that the max-norm of the low rank component $\|\boldsymbol{\Theta}_2\|_{\max}$ is less than $\frac{\alpha}{\sqrt{pk}}$ for some fixed $\alpha > 0$.

In this paper, we consider a general design matrix $\boldsymbol{A}$ and parameter matrices $(\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2)$, for regression analysis, where $\boldsymbol{A}$ represents features of observations which is deterministic, and can be any matrix with n rows and p columns. Of particular interest is reconstruction of $\boldsymbol{\Theta}$ in a high-dimensional situation in which (p, k) may exceed the sample size n. Computationally, we use an alternating direction method separating low-rank pursuit from sparsity pursuit alternatively, where an exact solution to the low-rank problem and that to the sparsity pursuit problem when $\boldsymbol{A} = \boldsymbol{I}_{n\times p}$ or an approximated solution for a general $\boldsymbol{A}$ is obtained. In either case, the final solution is shown to be stationary without and with maximum block improvement (Chen et al., 2012) for $\boldsymbol{A} = \boldsymbol{I}_{n\times p}$ and a general $\boldsymbol{A}$. Theoretically, we establish error bound for the proposed method in the Hellinger-distance for reconstruction of $\boldsymbol{\Theta}$, based on which rates of convergence are obtained. Numerically, the proposed method compares favorably against two strong competitors in simulations.

The paper is organized as follows. Section 2 develops a computational method through the alternating directions method and a closed-form solution for a rank problem. Section 3 investigates statistical properties of the proposed method, followed by simulation studies and a real data example in Section 4. Finally, technical proofs are contained in Section 5.

2. Proposed Method

In this section, we explore a structure decomposition of a parameter matrix in the form $\boldsymbol{\Theta} = \boldsymbol{\Theta}_1 + \boldsymbol{\Theta}_2$ under model (1), then develops computational methods in two situations and discuss their properties.

2.1 Structure Decomposition

Due to non-uniqueness of such a decomposition under model (1), we seek one decomposition, among many overcomplete decompositions, that minimizes the effective degrees of freedom

of $\boldsymbol{\Theta}$ Efron (2004), defined as

$$\mathrm{Eff}(\boldsymbol{\Theta}) = \min_{\{\boldsymbol{\Theta}=\boldsymbol{\Theta}_1+\boldsymbol{\Theta}_2:\|\boldsymbol{\Theta}_1\|_0\le \max(0,p+k-2r(\boldsymbol{\Theta}_2)-2)\}} \|\boldsymbol{\Theta}_1\|_0 + (p+k-r(\boldsymbol{\Theta}_2))r(\boldsymbol{\Theta}_2),$$

where $\|\cdot\|_0$ is the L_0-norm of a matrix, or the number of nonzero entries of the matrix, and $r(\cdot)$ denotes the rank of a matrix. In other words, we identify a decomposition minimizing the effective degrees of freedom $\mathrm{Eff}(\boldsymbol{\Theta})$, among all candidate decompositions. Lemma 1 below says that the minimal of $\mathrm{Eff}(\boldsymbol{\Theta})$ is unique in $(\|\boldsymbol{\Theta}_1\|_0, r(\boldsymbol{\Theta}_2))$ under the constraint that $\|\boldsymbol{\Theta}_1\|_0 \le \max(0, p+k-2r(\boldsymbol{\Theta}_2)-2) \le 2\max(p,k)$.

Lemma 1 *The minimizer of Eff*$(\boldsymbol{\Theta})$ *is unique with respect to* $(\|\boldsymbol{\Theta}_1\|_0, r(\boldsymbol{\Theta}_2))$ *if* $\|\boldsymbol{\Theta}_1\|_0 \le \max(0, p+k-2r(\boldsymbol{\Theta}_2)-2)$. *Moreover,*

$$\textit{Eff}(\boldsymbol{\Theta}) \le \min((p+k-r(\boldsymbol{\Theta}))r(\boldsymbol{\Theta}), \|\boldsymbol{\Theta}\|_0)).$$

Model (1) is identifiable with respect to $\boldsymbol{\Theta}$ but may not be so in $(\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2)$ even when $\boldsymbol{A}$ is of full rank, due to non-uniqueness of a decomposition $\boldsymbol{\Theta} = \boldsymbol{\Theta}_1 + \boldsymbol{\Theta}_2$.

2.2 Estimation

To pursue structures of low-rank and sparsity through matrix decomposition simultaneously, we propose a constrained likelihood method subject to two nonconvex constraints:

$$\min_{\boldsymbol{\Theta}_1,\boldsymbol{\Theta}_2} \|\boldsymbol{A\Theta_1} + \boldsymbol{A\Theta_2} - \boldsymbol{Z}\|_F^2, \quad \text{subject to} \quad \|\boldsymbol{\Theta}_1\|_0 \le s_1, \quad r(\boldsymbol{\Theta}_2) \le s_2, \tag{3}$$

where $\|\cdot\|_F$ is the Frobenius-norm defined as the L_2-norm of all entries of a matrix, and s_1 and s_2 are integer-valued tuning parameters with $0 \le s_1 \le \max(p,k)$ and $1 \le s_2 \le \min(n,k,p)$ based on the consideration that the rank function and the sparsity measure are integer-valued.

When $\boldsymbol{A} = \boldsymbol{I}_{n\times p}$, (3) is simplified as

$$\min_{\boldsymbol{\Theta}_1,\boldsymbol{\Theta}_2} \|\boldsymbol{Z} - \boldsymbol{\Theta}_1 - \boldsymbol{\Theta}_2\|_F^2 \quad \text{subject to } \|\boldsymbol{\Theta}_1\|_0 \le s_1, \quad r(\boldsymbol{\Theta}_2) \le s_2, \tag{4}$$

where a special structure may be taken into account to solve this nonconvex minimization.

When $\boldsymbol{A} \ne \boldsymbol{I}_{n\times p}$ is any matrix of full rank, the two constraints in (3) are either defined by the L_0-function or the rank function, imposing computational challenges. To develop an efficient algorithm to solve (3), we approximate the $\|\boldsymbol{\Theta}_1\|_0 = \sum_{i,j} I(|\theta_{ij}| \ne 0)$ by its computational surrogate—the truncated L_1-function $\sum_{\theta_{ij}\in\boldsymbol{\Theta}_1} \frac{1}{\tau}\min(|\theta_{ij}|, \tau)$ Shen et al. (2012) as $\tau \to 0^+$. This leads to a computational surrogate of (3):

$$\min_{\boldsymbol{\Theta}_1,\boldsymbol{\Theta}_2} f(\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2), \text{ subject to } \frac{1}{\tau}\sum_{i,j}\min(|\theta_{ij}|,\tau) \le s_1, \ r(\boldsymbol{\Theta}_2) \le s_2, \tag{5}$$

where $f(\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2) = \|\boldsymbol{A}(\boldsymbol{\Theta}_1 + \boldsymbol{\Theta}_2) - \boldsymbol{Z}\|_F^2$ and τ is a nonnegative tuning parameter.

2.3 Method for Nonconvex Minimization

This section will develop computational strategies for (4) and (5) separately, based on blockwise coordinate decent as well as maximum block improvement (MBI, Chen et al., 2012). First, we separate the task of sparsity pursuit for $\boldsymbol{\Theta}_1$ from that of rank minimization for $\boldsymbol{\Theta}_2$, where $\boldsymbol{\Theta}_1$ and $\boldsymbol{\Theta}_2$ correspond to two blocks for decent. Second, we apply MBI to assure that blockwise coordinate decent yields a stationary solution for nonconvex minimization, which would be otherwise impossible. In addition, for (5), we develop a gradient project method to permit fast computation of a constrained problem through the means of unconstrained optimization.

The strategy of blockwise coordinate decent proceeds as follows. For (4) and (5), we solve it in $\boldsymbol{\Theta}_2$ given $\boldsymbol{\Theta}_1$ and solve them in $\boldsymbol{\Theta}_1$ given $\boldsymbol{\Theta}_2$, alternatively. In each step of alternating blocks, we proceed with the block giving the maximum block improvement.

2.3.1 Nonconvex minimization (4): a special case

For (4), when $\boldsymbol{\Theta}_2$ is held fixed, (4) has a global minimizer can be obtained through componentwise thresholding defined by the L_0-function as follows:

$$\hat{\boldsymbol{\Theta}}_1(\boldsymbol{Z}, \boldsymbol{\Theta}_2) = \left(I\left\{ |z_{ij} - \theta_{ij}^{(2)}| > \lambda \right\} \cdot (z_{ij} - \theta_{ij}^{(2)}) \right)_{p\times k}, \tag{6}$$

where $\theta_{ij}^{(2)}$ is the ijth entry of $\boldsymbol{\Theta}_2$ and λ is any number between the s_1th and (s_1+1)th largest entries of $|\boldsymbol{Z} - \boldsymbol{\Theta}_2|$.

When $\boldsymbol{\Theta}_1$ is held fixed, a global minimizer of (4) is

$$\hat{\boldsymbol{\Theta}}_2(\boldsymbol{Z}, \boldsymbol{\Theta}_1) = \boldsymbol{U}\boldsymbol{D}_{s_2}\boldsymbol{V}^T, \tag{7}$$

where $\boldsymbol{U}$ and $\boldsymbol{V}$ are given by singular value decomposition (SVD) of $\boldsymbol{Z} - \boldsymbol{\Theta}_1 = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T$ and $\boldsymbol{D}_{s_2}$ is a diagonal matrix retaining the largest s_2 singular values of $\boldsymbol{Z} - \boldsymbol{\Theta}_1$ and truncating other singular values at zero.

Our algorithm for computing (4) is summarized.

Step 1.(Initialization) Supply a good initial estimate $(\hat{\boldsymbol{\Theta}}_1^{(0)}, \hat{\boldsymbol{\Theta}}_2^{(0)})$ in (4). Specify precision $\delta > 0$.

Step 2.(Iteration) At iteration m, update $\hat{\boldsymbol{\Theta}}_2^{(m)}$ in (7) with $\boldsymbol{\Theta}_1 = \hat{\boldsymbol{\Theta}}_1^{(m-1)}$. Then update $\hat{\boldsymbol{\Theta}}_1^{(m)}$ in (6) with $\boldsymbol{\Theta}_2 = \hat{\boldsymbol{\Theta}}_2^{(m)}$.

Step 3.(Stopping rule) Terminate if $|f(\hat{\boldsymbol{\Theta}}_1^{(m)}, \hat{\boldsymbol{\Theta}}_2^{(m)}) - f(\hat{\boldsymbol{\Theta}}_1^{(m-1)}, \hat{\boldsymbol{\Theta}}_2^{(m-1)})| \le \delta$, where $f(\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2) = \|\boldsymbol{\Theta}_1 + \boldsymbol{\Theta}_2 - \boldsymbol{Z}\|_F^2$. Let m^* be the index at termination. The estimate is then $(\hat{\boldsymbol{\Theta}}_1^{(m^*)}, \hat{\boldsymbol{\Theta}}_2^{(m^*)})$.

2.3.2 Nonconvex minimization (5): A general case

The problem of solving for $\boldsymbol{\Theta}_2$ in (5) given $\boldsymbol{\Theta}_1$ reduces to that of constrained rank minimization

$$\min_{\boldsymbol{\Theta}_2} \|\boldsymbol{A}\boldsymbol{\Theta}_2 - (\boldsymbol{Z} - \boldsymbol{A}\boldsymbol{\Theta}_1)\|_F^2 \quad \text{subject to} \quad r(\boldsymbol{\Theta}_2) \le s_2, \tag{8}$$

provided that $\boldsymbol{\Theta}_1$ satisfies the sparsity constraint in (5). Now write $\boldsymbol{\Theta}_2 \equiv \boldsymbol{CF}$, where $\boldsymbol{C}$ and $\boldsymbol{F}$ are $p \times r$ and $r \times k$ matrices with $r \leq s_2$, consisting of a basis of the column space and that of the row space of $\boldsymbol{\Theta}_2$, respectively. Note that $\{\boldsymbol{\Theta}_2 : r(\boldsymbol{\Theta}_2) \leq s_2\} = \{\boldsymbol{\Theta}_2 : \boldsymbol{\Theta}_2 = \boldsymbol{CF}, r \leq s_2\}$. Then solving (8) is equivalent to that

$$\min_{\boldsymbol{C},\boldsymbol{F}} \|\boldsymbol{A}(\boldsymbol{CF}) - (\boldsymbol{Z} - \boldsymbol{A\Theta}_1)\|_F^2, \tag{9}$$

An application of an argument of (Xing et al., 2012) yields a global minimizer of (9), which has an analytic form

$$\hat{\boldsymbol{\Theta}}_2(\boldsymbol{\Theta}_1) = \hat{\boldsymbol{C}}\hat{\boldsymbol{F}}, \quad \hat{\boldsymbol{C}} = \boldsymbol{V}\boldsymbol{D}^{-1}\boldsymbol{U}_w, \quad \hat{\boldsymbol{F}} = \boldsymbol{D}_w\boldsymbol{V}_w^T, \tag{10}$$

where $\boldsymbol{D}$ is a $r(\boldsymbol{A}) \times r(\boldsymbol{A})$ diagonal singular vector matrix based on SVD of $\boldsymbol{A} = \boldsymbol{UDV}^T$, $\boldsymbol{D}_w$ is also a diagonal matrix of s_2 leading singular values of $\boldsymbol{W} \equiv \boldsymbol{U}^T(\boldsymbol{Z} - \boldsymbol{A\Theta}_1)$ and U_w, V_w are matrices consisting of the corresponding right and left singular vectors.

Note that computation involves only the first s_2 largest singular values. Therefore, we employ the randomized truncated SVD method (Halko et al., 2011), for efficient computation of a large problem. This amounts to a complexity of order $O(pk \log r)$, as compared to $O(\min(pk^2, p^2k))$ of a conventional SVD method (Golub & Van, 1996).

Solving for $\boldsymbol{\Theta}_1$ in (5) given $\boldsymbol{\Theta}_2$, on the other hand, becomes the problem of sparsity pursuit. In particular, we solve, assuming that $r(\boldsymbol{\Theta}_2) \leq s_2$,

$$\min_{\boldsymbol{\Theta}_1} \|\boldsymbol{A\Theta}_1 - (\boldsymbol{Z} - \boldsymbol{A\Theta}_2)\|_F^2, \quad \text{subject to} \quad \frac{1}{\tau}\sum_{\theta_{ij} \in \boldsymbol{\Theta}_1} \min(|\boldsymbol{\theta}_{ij}|, \tau) \leq s_1, \tag{11}$$

which is solved iteratively by a difference of convex (DC) programming, constructing a convex set containing the original constrained set. The constraint in (5) is defined by $J(\boldsymbol{\Theta}_1) = S_1(\boldsymbol{\Theta}_1) - S_2(\boldsymbol{\Theta}_1)$ with $S_1(\boldsymbol{\Theta}_1) = \frac{1}{\tau}\sum|\theta_{ij}|$ and $S_2(\boldsymbol{\Theta}_1) = \frac{1}{\tau}\sum\max(|\theta_{ij}| - \tau, 0)$ are convex in $\boldsymbol{\Theta}_1$. Then a sequence of upper approximations of $J(\boldsymbol{\Theta}_1)$ is constructed: At iteration step m by $J^{(m)}(\boldsymbol{\Theta}_1) = \sum_{\theta_{ij} \in \boldsymbol{\Theta}_1}\left(\frac{|\theta_{ij}|}{\tau} I(|\hat{\theta}_{ij}^{(m-1)}| \leq \tau) + I(|\hat{\theta}_{ij}^{(m-1)}| > \tau)\right)$. This yields a sequence of convex minimization subproblems with convex constraints: At iteration step m, we solve

$$\min_{\boldsymbol{\Theta}_1} \|\boldsymbol{A\Theta}_1 - (\boldsymbol{Z} - \boldsymbol{A\Theta}_2)\|_F^2, \quad \text{subject to} \quad J^{(m)}(\boldsymbol{\Theta}_1) \leq s_1. \tag{12}$$

For (12), we develop a gradient projection method. First, we generalize an l_1-ball result of (Liu & Ye, 2009) to (12).

Lemma 2 *(Projection) For any set $K \subseteq \{1, 2, \cdots, n\}$,*

$$\boldsymbol{x}^* = \mathcal{T}_{K,z}(\boldsymbol{v}) = \operatorname*{argmin}_{\boldsymbol{x} \in \mathbb{R}^n : \sum_{i \in K}|x_i| \leq z} \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{v}\|_2^2,$$

where $\mathcal{T}_{K,z} : \mathbb{R}^n \to \mathbb{R}^n$ is a projection operator defined by

$$\mathcal{T}_{K,z}(\boldsymbol{v})_i = sign(v_i)\max(|v_i| - \lambda^*, 0)$$

where $\lambda^* = 0$ if $\sum_{i \in K}|v_i| \leq z$ or $i \notin K$ and $\lambda^* = \frac{\sum_{i \in K \setminus K_0}|v_i| - z}{|K| - |K_0|}$ otherwise, and $K_0 = \{j : \sum_{i \in K}\max(|v_i| - |v_j|, 0) - z > 0\}$.

Before solving (12), we simply extend the fast iterative shrinkage-thresholding (FISTA) algorithm (Beck & Teboulle, 2009) to solving (13).

Lemma 3 *For any set K defined in Lemma 2, a global minimizer of*

$$\min_{\boldsymbol{x}\in\mathbb{R}^n:\sum_{i\in K}|x_i|\le z} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2^2 \tag{13}$$

can be obtained by FISTA iteratively: At iteration step t:

$$\begin{aligned}
\boldsymbol{x}^{(t)} &= \mathcal{T}_{K,z}\Big(\boldsymbol{y}^{(t)} - \frac{1}{2L}\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{y}^{(t)}-\boldsymbol{b})\Big),\\
\rho_{t+1} &= \frac{1+\sqrt{1+4\rho_t^2}}{2},\\
\boldsymbol{y}^{(t+1)} &= \boldsymbol{x}^{(t)} + \left(\frac{\rho_t-1}{\rho_{t+1}}\right)(\boldsymbol{x}^{(t)}-\boldsymbol{x}^{(k-1)}),
\end{aligned}$$

where L is the largest singular value of $\boldsymbol{A}$.

Next we solve (12) using Lemma 3, which yields an analytic updating formula in a matrix form.

Then a global minimizer of (12) is computed using an iterative scheme with respect to t as follows:

$$\begin{aligned}
&\boldsymbol{v}^{(1)} = \hat{\boldsymbol{\Theta}}_1^{(m,0)} = \hat{\boldsymbol{\Theta}}_1^{(m-1)}, \quad \rho_1 = 1,\\
&\hat{\boldsymbol{\Theta}}_1^{(m,t)} = \mathcal{T}_{K^{(m)},z^{(m)}}\left(\boldsymbol{v}^{(t)} - \frac{1}{2\lambda_{\max}(\boldsymbol{A}^{\boldsymbol{T}}\boldsymbol{A})}\boldsymbol{A}^T[\boldsymbol{A}\boldsymbol{v}^{(t)} - (\boldsymbol{Z}-\boldsymbol{A}\boldsymbol{\Theta}_2)]\right),\\
&\rho_{t+1} = \frac{1+\sqrt{1+4\rho_t^2}}{2}, \quad \boldsymbol{v}^{(t+1)} = \hat{\boldsymbol{\Theta}}_1^{(m,t)} + \left(\frac{\rho_t-1}{\rho_{t+1}}\right)(\hat{\boldsymbol{\Theta}}_1^{(m,t)} - \hat{\boldsymbol{\Theta}}_1^{(m,t-1)}),
\end{aligned} \tag{14}$$

where $K^{(m)} = \{(i,j) : |\hat{\theta}_{ij}^{(m-1)}| \le \tau\}$, $z^{(m)} = \tau(s_1 - \sum_{\theta_{ij}\in\boldsymbol{\Theta}_1} I(|\hat{\theta}_{ij}^{(m-1)}| > \tau))$ and $\lambda_{\max}(\cdot)$ denotes the largest eigenvalue of a matrix.

The algorithm is summarized as follows.

Algorithm 2:

Step 1.(Initialization) Supply a good initial estimate $(\hat{\boldsymbol{\Theta}}_1^{(0)}, \hat{\boldsymbol{\Theta}}_2^{(0)})$ in (5). Specify precision $\delta > 0$.

Step 2.(Iteration) At iteration m, compute candidate $\hat{\boldsymbol{\Theta}}_2$ in (10) with $\boldsymbol{\Theta}_1 = \hat{\boldsymbol{\Theta}}_1^{(m-1)}$ and candidate $\hat{\theta}_{ij} \in \hat{\boldsymbol{\Theta}}_1$ in (14) with $\boldsymbol{A}\boldsymbol{\Theta}_2 = \boldsymbol{A}\hat{\boldsymbol{\Theta}}_2^{(m-1)}$.

Step 3.(Maximum block improvement) At each iteration m, determine which of the two candidates $(\hat{\boldsymbol{\Theta}}_1, \hat{\boldsymbol{\Theta}}_2^{(m-1)})$ and $(\hat{\boldsymbol{\Theta}}_1^{(m-1)}, \hat{\boldsymbol{\Theta}}_2)$ for updating according to the amounts of improvement. That is, update $(\hat{\boldsymbol{\Theta}}_1^{(m)}, \hat{\boldsymbol{\Theta}}_2^{(m)}) = (\hat{\boldsymbol{\Theta}}_1, \hat{\boldsymbol{\Theta}}_2^{(m-1)})$ if $f(\hat{\boldsymbol{\Theta}}_1, \hat{\boldsymbol{\Theta}}_2^{(m-1)}) \le f(\hat{\boldsymbol{\Theta}}_1^{(m-1)}, \hat{\boldsymbol{\Theta}}_2)$; update $(\hat{\boldsymbol{\Theta}}_1^{(m)}, \hat{\boldsymbol{\Theta}}_2^{(m)}) = (\hat{\boldsymbol{\Theta}}_1^{(m-1)}, \hat{\boldsymbol{\Theta}}_2)$ otherwise.

Step 4.(Stopping rule) Terminate if $|f(\hat{\boldsymbol{\Theta}}_1^{(m)}, \hat{\boldsymbol{\Theta}}_2^{(m)}) - f(\hat{\boldsymbol{\Theta}}_1^{(m-1)}, \hat{\boldsymbol{\Theta}}_2^{(m-1)})| \le \delta$. Denote by m^* the index at termination. The final estimate is

$$\hat{\boldsymbol{\Theta}}_1 = \hat{\boldsymbol{\Theta}}_1^{(m^*)}, \quad \hat{\boldsymbol{\Theta}}_2 = \hat{\boldsymbol{C}}\hat{\boldsymbol{F}},$$

where $\hat{\boldsymbol{C}}$ and $\hat{\boldsymbol{F}}$ are defined in (10) with $\boldsymbol{\Theta}_1 = \hat{\boldsymbol{\Theta}}_1$.

2.4 Computational Properties

This section discusses computational properties of Algorithms 1 and 2. For nonconvex minimization, our methods may not guarantee a global minimizer for (3). However, the following lemma says that our solution of Algorithms 1 and 2 yields a stationary point of the cost function. Note that the scheme of maximum block improvement is essential for the result of Lemma 5.

Lemma 4 *The minimal cost function $f(\hat{\boldsymbol{\Theta}}_1^{(m)}, \hat{\boldsymbol{\Theta}}_2^{(m)})$ in Algorithm 1 is strictly decreasing in m before termination. Moreover, the solution is a stationary point of $f(\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2)$ in that $\theta_{ij}^{(*)} = \text{argmin}_{\theta_{ij} \in \boldsymbol{\Theta}_k; k=1,2} f((\boldsymbol{\Theta}_1^*, \boldsymbol{\Theta}_2^*) \setminus \theta_{ij})$, where $(\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2) \setminus \theta_{ij}$ is the set of parameters of $(\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2)$ without one component θ_{ij} in $\boldsymbol{\Theta}_1$ or $\boldsymbol{\Theta}_2$, and $(\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2)$ satisfy the constraints in* (5).

Lemma 5 *If $\boldsymbol{A}$ is of full rank, then $\hat{\boldsymbol{\Theta}}_1$ computed from* **Algorithm 2** *satisfies the constraints in* (12). *Moreover, the minimal cost function $f(\hat{\boldsymbol{\Theta}}_1^{(m)}, \hat{\boldsymbol{\Theta}}_2^{(m)})$ is strictly decreasing in m before termination. Finally, if the solution $(\hat{\boldsymbol{\Theta}}_1, \hat{\boldsymbol{\Theta}}_2)$ satisfies* (5) *and it is a stationary point of $f(\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2)$ in that*

$$\theta_{ij}^{(*)} = \underset{\theta_{ij} \in \boldsymbol{\Theta}_k; k=1,2}{\text{argmin}} f((\boldsymbol{\Theta}_1^*, \boldsymbol{\Theta}_2^*) \setminus \theta_{ij}),$$

where $(\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2) \setminus \theta_{ij}$ is the set of parameters of $(\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2)$ without one component θ_{ij} in $\boldsymbol{\Theta}_1$ or $\boldsymbol{\Theta}_2$, and $(\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2)$ satisfy the constraints in (5).

With regard to the computational complexity of Algorithms 1 and 2, the method of truncated SVD yields an approximated SVD with a complexity of $O(pk \log r + (p+k)r^2)$ operations (Halko et al., 2011). Sorting requires a complexity of $O(pk \log(pk))$. For FISTA, the convergence rate is $O(1/t^2)$ (Beck & Teboulle, 2009), where t is the number of iterations. Overall, the computational complexity of Algorithm 1 is $O(pk \log(pk) + (p+k)r^2)I_2$, while that of Algorithm 2 is $O((pk \log r + (p+k)r^2 + I_1/\varepsilon^2)I_2$, where ε denotes the precision specified in Algorithm 2, and I_1 and I_2 is the number of DC iteration and blockwise iteration, respectively. Based on our experience, I_1 and I_2 are about between 3 and 20.

3. Theory

This section drives a finite-sample probability error bound for reconstruction of the true $\boldsymbol{\Theta}^0$ by $\hat{\boldsymbol{\Theta}}^{L_0}$, which is a global minimizer of (3) in that $\hat{\boldsymbol{\Theta}}^{L_0} = \hat{\boldsymbol{\Theta}}_1^{L_0} + \hat{\boldsymbol{\Theta}}_2^{L_0}$. Note that existence of a global minimizer is assured by the fact that the cost function (3) is bounded blow by zero. Moreover, we will provide an insight into simultaneous pursuit of the low rank and sparsity structures through matrix decomposition by contrasting the proposed method with (s_1, s_2) against low rank approximation alone with $(s_1 = 0, s_2)$ and sparsity pursuit alone with $(s_1, s_2 = 0)$.

Let $\|\boldsymbol{\Theta}\|_\infty = \max_i \sum_j |\theta_{ij}|$ and $\|\boldsymbol{\Theta}\|_{\max} = \max_{ij} |\theta_{ij}|$ are the L_∞-norm and max norm respectively. Before proceeding, we define a parameter space Λ as $\{\boldsymbol{\Theta} = \boldsymbol{\Theta}_1 + \boldsymbol{\Theta}_2 : \|\boldsymbol{\Theta}_1\|_0 \leq s_1, \|\boldsymbol{\Theta}_1\|_{\max} \leq l_1, \boldsymbol{\Theta}_2 = \boldsymbol{C}\boldsymbol{F}, \max(\|\boldsymbol{C}\|_\infty, \|\boldsymbol{F}^T\|_\infty) \leq l_2\}$, where $l_1, l_2 > 0$ are constant, $\boldsymbol{C}$ is a $p \times s_2$ matrix, $\boldsymbol{F}$ is a $s_2 \times k$ matrix, $\boldsymbol{F}^T$ is the transport of $\boldsymbol{F}$ and $s_2 > 0$ is an upper

bound of $r(\mathbf{\Theta}_2)$. Let $g(\mathbf{\Theta}, \boldsymbol{Z})$ be the probability density of $\boldsymbol{Z}$ with respect to dominating measure ν on Λ. Define the Hellinger distance between two densities as

$$h(\mathbf{\Theta}, \mathbf{\Theta}') = \frac{1}{2}\left(\int (g^{1/2}(\mathbf{\Theta}, \boldsymbol{Z}) - g^{1/2}(\mathbf{\Theta}', \boldsymbol{Z}))^2 d\nu\right)^{1/2}, \tag{15}$$

which will be used to measure estimation accuracy.

The following technical assumptions are made.

Assumption A: (Norm-relation) For any $\mathbf{\Theta}, \mathbf{\Theta}' \in \Lambda$ and any $\delta > 0$,

$$\int \sup_{\|\mathbf{\Theta}-\mathbf{\Theta}'\|_{\max}\le\delta} (g^{1/2}(\mathbf{\Theta}, y) - g^{1/2}(\mathbf{\Theta}', y))^2 d\nu(y) \le M^2\delta^2,$$

where M might depend on $p,\ k,\ s_1,\ s_2$ and l_1, l_2.

Assumption A specifies a norm relation between the metric $\|\cdot\|_{\max}$ over parameters and the Hellinger distance over the corresponding densities. This can be verified given a specific form of g.

Theorem 1 gives a probability error bound for $\hat{\mathbf{\Theta}}^{L_0}$ under probability P under the true $\mathbf{\Theta}^0$. Let (s_1^0, s_2^0) be the degree of sparsity and rank, as defined in $\mathrm{Eff}(\mathbf{\Theta}^0)$ in Lemma 1.

Theorem 6 *Under **Assumptions A**, for any $\epsilon \ge \epsilon_{n,p,k}$*

$$P\left(h(\hat{\mathbf{\Theta}}^{L_0}, \mathbf{\Theta}^0) \ge \epsilon\right) \le 5\exp(-c_1 n\epsilon^2),$$

$\epsilon_{n,p,k} = \frac{C_{p,k}}{\sqrt{n}}\sqrt{\log(\frac{\sqrt{n}}{C_{p,k}})}$ *with*

$$C_{p,k} = c_2\sqrt{\log(2^9 M c_4(l_2^3 + l_1))}\sqrt{(p+k)s_2^0 + s_1^0} + c_2\sqrt{s_1^0 \log\left(e\frac{(p+k-r(\mathbf{\Theta}^0))r(\mathbf{\Theta}^0)}{s_1^0}\right)}. \tag{16}$$

If $\log(r(\mathbf{\Theta}^0)) \le d s_2^0$ for some $d > 0$, then it can be simplified:

$$C_{p,k} = c_3\sqrt{\log(M)}\sqrt{(p+k-s_2^0)s_2^0},$$

where $c_1 - c_3$ are positive constants and M is defined in Assumption A. Moreover, as $n, p, k \to \infty$, $h^2(\hat{\mathbf{\Theta}}^{L_0}, \mathbf{\Theta}^0) = O_p(\epsilon_{n,p,k}^2)$, and $Eh^2(\hat{\mathbf{\Theta}}^{L_0}, \mathbf{\Theta}^0) = O(\epsilon_{n,p,k}^2)$, where $O_p(\cdot)$ and E denote the stochastic order and the expectation under P.

Corollary 1 gives an order of $\epsilon_{n,p,k}$ in three extreme situations with M held fixed.

Corollary 1 *Suppose M in Assumptions A is a constant independent of (p, k, s_1, s_2).*

(i) When $\mathbf{\Theta}^0$ is extremely sparse, that is, $\|\mathbf{\Theta}^0\|_0 \le p + k - 2$, $C_{p,k}$ in (16) is no worse than $O\left(\sqrt{\|\mathbf{\Theta}^0\|_0 \log((p+k-r(\mathbf{\Theta}^0))r(\mathbf{\Theta}^0)/\|\mathbf{\Theta}^0\|_0)}\right)$.

(ii) When $\mathbf{\Theta}^0$ is a low-rank matrix, $C_{p,k}$ in (16) is no worse than $O\left(\sqrt{(p+k-r(\mathbf{\Theta}^0))r(\mathbf{\Theta}^0)}\right)$.

(iii) When $\mathbf{\Theta}^0$ *is dense, say* $\|\mathbf{\Theta}^0\|_0 \geq cpk$ *for a constant* $0 < c \leq 1$*, and of full rank,* $C_{p,k}$ *in* (16) *is* $O\left(\max\left(\sqrt{(p+k-s_2^0)s_2^0}, \sqrt{s_1^0 \log(\frac{pk}{s_1^0})}\right)\right)$.

Then $C_{p,k}^L = O\left(\sqrt{(p+k-r(\mathbf{\Theta}^0))r(\mathbf{\Theta}^0)}\right)$.

Corollary 2 and Theorem 2 give a similar result under the Hellinger distance and the Kullback-Leibler distance, respectively, assuming that $\boldsymbol{\epsilon}_i$ follows a normal distribution.

Corollary 2 *If* $\boldsymbol{\epsilon}_i$ *in* (1) *follows* $N(0, \sigma^2 I_{k\times k})$*,* $\|\boldsymbol{A}\|_\infty$ *is bounded, then the results in Corollary 1 continue to hold.*

Theorem 7 *Under the same assumptions in Corollary 2, we have, for any* $\epsilon \geq \epsilon_{n,p,k}$*,*

$$P\left(K(\mathbf{\Theta}^0, \hat{\mathbf{\Theta}}^{L_0}) \geq 4\epsilon^2\right) \leq 5\exp(-c_1 n\epsilon^2).$$

where $K(\cdot,\cdot)$ *is Kullback-Leibler distance under normality and* $\epsilon_{n,p,k}$ *and* c_2 *remain to be the same as in Theorem 1. As* $n, p, k \to \infty$*,* $K(\mathbf{\Theta}^0, \hat{\mathbf{\Theta}}^{L_0}) = O_p(\epsilon_{n,p,k}^2)$ *and* $EK(\mathbf{\Theta}^0, \hat{\mathbf{\Theta}}^{L^0}) = O(\epsilon_{n,p,k}^2)$.

Theorem 3 gives an error bound for $\|\hat{\mathbf{\Theta}}^{L_0} - \mathbf{\Theta}^0\|_F^2$ under the normal assumption when $\boldsymbol{A} = \boldsymbol{I}_{n\times p}$.

Theorem 8 *Assume that* $\boldsymbol{A} = \boldsymbol{I}_{n\times p}$ *with* $n = \max(p,k)$*. Under the same assumptions in Corollary 2 with* $\sigma = O(\frac{1}{\sqrt{\max(p,k)}})$*, as* $n, p, k \to \infty$*,* $\|\hat{\mathbf{\Theta}}^{L_0} - \mathbf{\Theta}^0\|_F^2 = O_p(C'_{p,k}\log(\frac{1}{C'_{p,k}}))$*, where*

$$C'_{p,k} = \frac{\log(\max(p,k)) \cdot [(p+k)s_2^0 + s_1^0] + s_1^0 \log\left(e\frac{(p+k-r(\mathbf{\Theta}^0))r(\mathbf{\Theta}^0)}{s_1^0}\right)}{\max(p^2, k^2)}$$

4. Numerical Examples

This section examines operating characteristics of the proposed method through simulations, and demonstrates its effectiveness on applications in image reconstruction and in time series analysis. In the literature, it is known that the state-of-art methods are the low-rank approximation method subject to rank restriction as well as its regularized version, which outperforms the low-rank approximation method with the trace-norm (Xing et al., 2012; She, 2013; Zhou & Tao, 2011). In Section 4.1, we contrast our proposed method with pursuing low rank and sparsity structures through matrix decomposition simultaneously, with the former low rank approximation method subject to rank restriction (low-rank alone), as well as the method based on sparsity pursuit alone (sparsity alone). Here Algorithm 2 are used. Most importantly, in Section 4.2, we compare the proposed method using Algorithm 1 with two strong competitors the method of Go Decomposition (GoDec, Zhou & Tao, 2011) and the method augmented Lagrange multipliers (ALM, Lin et al., 2009) when $\boldsymbol{A} = \boldsymbol{I}_{n\times p}$ in (2). In simulations, codes for ALM and GoDec are used at the authors' website, and the initial values for Algorithms 1 and 2 are set to be the zero-matrix

4.1 Simulation I: Operating Characteristics

The simulated example is generated as follows. First, a $n \times p$ design matrix $\boldsymbol{A}$ is sampled with each entry being iid $N(0,1)$. Second, the true $\boldsymbol{\Theta}_1$ is a $p \times k$ matrix with all diagonals one and two more non-zeros (2 and 2) being randomly chosen with equal probability, and the true $\boldsymbol{\Theta}_2$ is generated by multiplying a $p \times r$ matrix with a $r \times k$ matrix with each entry following $N(1,1)$. Moreover, each entry of $\boldsymbol{E}$ is iid $N(0, 0.25)$. Throughout the simulations, $\boldsymbol{\Theta}_1$ and $\boldsymbol{\Theta}_2$ are held fixed with different values of (n, p, k).

The proposed method is trained with a training set, and the optimal tuning parameters, minimizing the prediction mean squares error over an independent tuning set, are obtained through a bisection search over integer values. Then a method's performance is examined over a test set. The training, tuning and testing data sizes are n, $4n$ and $2n$.

For parameter estimation, we employ the mean squares error to evaluate performance

$$\frac{1}{4n}\|\boldsymbol{A}(\hat{\boldsymbol{\Theta}} - \boldsymbol{\Theta}^0)\|_F^2. \tag{17}$$

For rank recovery, we calculate the absolute difference between an estimated rank $\hat{r}$ and the true rank r_0, that is $|\hat{r} - r_0|$. For sparsity pursuit, we define the true positive (TP) as a ratio of the true positive numbers of nonzero estimates over the number of nonzeros in the true model, and the false positive (FP) as a ratio of the false positive numbers of nonzero estimates over the number of zeros in the true model. Here "Low rank alone", "Sparsity alone" and "Ours" indicate the low rank method subject to rank restriction, the sparsity pursuit method, and the proposed method

As indicated in Table 1, the proposed method performs favorably against its counterpart—the low rank approximation method subject to rank restriction and sparsity pursuit alone, across all situations with different values of n, p and k. Moreover, the proposed method enables to identify two structures through matrix decomposition simultaneously. In particular, it recovers the true rank of the matrix with nearly zero $|\hat{r} - r_0|$-values as compared to relatively large $|\hat{r} - r_0|$-values, ranging from 6.7 to 29.6, for its low-rank counterpart. At the same time, the proposed method has high true positives ranging from .92 to 1.00 and low false positives between 0.00 and 0.01, as compared to true positives ranging 0.04 to .44 and false positives between 0.03 and 0.20 of its counterpart based on sparsity pursuit. This suggests that pursuit of two types of structures is indeed advantageous than that of either one structure individually. This is mainly because these two structures are complementary to each other. As a result, higher parameter estimation accuracy, as measured by the MSE values, can be realized. In fact, the amount of improvement is large, which ranges from 147% to 1185400%. To see how each method performs as (n, p) increases, we fix $k = 5$.

As suggested by Table 2, the proposed method yields more stable performance than its two counterparts whose performance deteriorates rapidly, as the level of difficulty of a problem escalates when p and k increase.

4.2 Simulation II: Comparison

To compare with ALM (Lin et al., 2009) and GoDec (Zhou & Tao, 2011) for RPCA, consider the case of $\boldsymbol{A} = \boldsymbol{I}_{n\times p}$ in (2) and $p = k$ as in these papers. GoDec minimizes

$$\min_{\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2} \|\boldsymbol{Z} - \boldsymbol{\Theta}_1 - \boldsymbol{\Theta}_2\|_F^2 \quad \text{subject to } \text{card}(\boldsymbol{\Theta}_1) \le s_1,\ \text{rank}(\boldsymbol{\Theta}_2) \le s_2, \tag{18}$$

	$p=20,\ k=10$								
	Ours				Low-rank alone		Sparsity alone		
n	$\lvert\hat{r}-r_0\rvert$	TP	FP	MSE	$\lvert\hat{r}-r_0\rvert$	MSE	TP	FP	MSE
50	0.00	1.00	0.01	0.68	6.71	1.68	0.44	0.07	1367.03
	(0.00)	(0.00)	(0.03)	(0.15)	(0.52)	(0.28)	(0.29)	(0.01)	(173.50)
	$p=30,\ k=20$								
	Ours				Low-rank alone		Sparsity alone		
n	$\lvert\hat{r}-r_0\rvert$	TP	FP	MSE	$\lvert\hat{r}-r_0\rvert$	MSE	TP	FP	MSE
50	0.00	1.00	0.00	1.54	15.69	7.79	0.12	0.14	4650.35
	(0.00)	(0.00)	(0.00)	(0.30)	(2.41)	(1.03)	(0.21)	(0.02)	(511.55)
100	0.00	1.00	0.00	0.51	16.94	2.16	0.13	0.05	4399.38
	(0.00)	(0.00)	(0.00)	(0.08)	(0.24)	(0.18)	(0.22)	(0.01)	(429.41)
	$p=20,\ k=30$								
	Ours				Low-rank alone		Sparsity alone		
n	$\lvert\hat{r}-r_0\rvert$	TP	FP	MSE	$\lvert\hat{r}-r_0\rvert$	MSE	TP	FP	MSE
50	0.00	1.00	0.00	1.06	16.66	5.06	0.43	0.06	4276.25
	(0.00)	(0.00)	(0.00)	(0.17)	(0.76)	(0.62)	(0.28)	(0.01)	(508.06)
100	0.00	1.00	0.00	0.46	16.99	1.88	0.53	0.06	4087.58
	(0.00)	(0.00)	(0.00)	(0.05)	(0.10)	(0.16)	(0.20)	(0.01)	(406.97)
	$p=40,\ k=30$								
	Ours				Low-rank alone		Sparsity alone		
n	$\lvert\hat{r}-r_0\rvert$	TP	FP	MSE	$\lvert\hat{r}-r_0\rvert$	MSE	TP	FP	MSE
50	0.00	1.00	0.00	4.08	1.88	19.39	0.09	0.20	12018.68
	(0.00)	(0.00)	(0.00)	(1.21)	(0.59)	(1.57)	(0.20)	(0.04)	(1422.84)
	$p=50,\ k=20$								
	Ours				Low-rank alone		Sparsity alone		
n	$\lvert\hat{r}-r_0\rvert$	TP	FP	MSE	$\lvert\hat{r}-r_0\rvert$	MSE	TP	FP	MSE
100	0.00	1.00	0.00	0.95	16.86	5.05	0.04	0.03	11262.97
	(0.00)	(0.00)	(0.00)	(0.15)	(0.35)	(0.40)	(0.14)	(0.01)	(1003.69)
	$p=200,\ k=100$								
	Ours				Low-rank alone		Sparsity alone		
n	$\lvert\hat{r}-r_0\rvert$	TP	FP	MSE	$\lvert\hat{r}-r_0\rvert$	MSE			
300	3.76	0.92	0.00	8.26	29.56	54.24	–	–	–
	(1.24)	(0.23)	(0.00)	(0.86)	(7.84)	(0.81)	(–)	(–)	(–)

Table 1: Results of Simulation I. Algorithm 2 is used for computation.

where card$(\cdot)$ denotes the cardinality, and $s_j \geq 0$ are tuning parameters as in our case. Similarly, ALM that focuses on the non-noisy situation minimizes

$$\min_{\boldsymbol{\Theta}_1,\boldsymbol{\Theta}_2} \|\boldsymbol{\Theta}_2\|_* + \lambda \sum_{\theta_{ij}\in\boldsymbol{\Theta}_1} |\theta_{ij}|, \quad \text{subject to } \boldsymbol{Z} = \boldsymbol{\Theta}_1 + \boldsymbol{\Theta}_2, \tag{19}$$

where $\|\cdot\|_*$ is the nuclear-norm of a matrix.

Our simulation example remains the same as before except that the positions of nonzero elements in $\boldsymbol{\Theta}_2$ are randomly sampled with equal probability, in particular, $.1p$ and $.3p$ nonzeros are randomly chosen without replacement. For tuning, grid search is employed for GoDec in (18), with $1 \leq s_1 \leq (p+k)$ and $1 \leq s_2 \leq \min(p, k, 50)$; λ is fixed at $\frac{1}{\sqrt{p}}$ for (19).

		Ours				Low-rank alone		Sparsity alone		
n	p	$\lvert\hat{r}-r_0\rvert$	TP	FP	MSE	$\lvert\hat{r}-r_0\rvert$	MSE	TP	FP	MSE
50	20	0.00	1.00	0.002	0.58	2.00	0.84	0.433	0.08	570
		(0.00)	(0.00)	(0.006)	(0.14)	(0.00)	(0.18)	(0.30)	(0.02)	(73)
50	30	0.00	0.57	0.01	1.29	1.97	1.98	0.18	0.08	3772.33
		(0.00)	(0.17)	(0.01)	(0.32)	(0.17)	(0.42)	(0.27)	(0.01)	(542.38)
50	40	0.00	1.00	0.001	3.57	1.67	5.43	0.07	0.05	1998
		(0.00)	(0.00)	(0.003)	(1.58)	(0.60)	(1.73)	(0.18)	(0.01)	(257)
50	50	0.82	0.36	0.01	487.43	0.82	12255	0.05	0.03	3797
		(0.84)	(0.38)	(0.01)	(1081.68)	(0.81)	(79570)	(0.15)	(0.01)	(539)
100	20	0.00	1.00	0.01	0.23	2.00	0.32	0.53	0.08	541
		(0.00)	(0.00)	(0.03)	(0.05)	(0.00)	(0.05)	(0.21)	(0.02)	(58)
100	30	0.00	0.71	0.01	0.36	2.00	0.54	0.19	0.03	1461
		(0.00)	(0.25)	(0.01)	(0.05)	(0.00)	(0.08)	(0.21)	(0.01)	(147)
100	40	0.00	0.98	0.01	0.53	2.00	0.83	0.10	0.03	1929
		(0.00)	(0.10)	(0.02)	(0.08)	(0.00)	(0.11)	(0.20)	(0.01)	(179)

Table 2: Results for Simulation I with fixed $k = 5$. Algorithm 2 is used for computation.

From Tables 3, it is evidenced that the proposed method outperforms ALM uniformly in terms of the MSE while being comparable to GoDec, in all the situations with different values of (p, k, σ). Moreover, it always recovers the true rank of the matrix perfectly with $|\hat{r} - r_0| = 0$. Although ALM has comparable high TP values, its FP values are high as well in that they are at least 0.6488. As a result, ALM never captures the true rank.

4.3 AR Face Database 20pt Markup

For face image reconstruction, we use a subset of *AR Face Data* for this experiment. The original image is available at `http://www-prima.inrialpes.fr/FGnet/data/05-ARFace/markup_large.png`, which is a colored one with size of $186 \times 200 \times 3$. To enable detailed testing, the image has been labeled with 20 facial features on the face. We convert the image into black and white and reduce it to size 171×180. The target image is displayed in Figure 1.

Figure 1: The converted AR face image with markup points.

Twenty one markup points around eyes, nose, mouth and cheeks, which are used to test face recognition or verification performance when the exact location of the face and features

nonzeros	p	k	σ	Method	$\lvert\hat{r}-r_0\rvert$	TP	FP	MSE
$0.1p$	50	30	0.1	Ours	0.0000	0.9940	0.0000	0.2366
					(0.0000)	(0.0343)	(0.0002)	(0.0251)
				ALM	13.0300	1.000	0.6488	1.5057
					(0.6735)	(0.0000)	(0.0082)	(0.0576)
				GoDec	0.0000	0.9940	0.0000	0.2363
					(0.0000)	(0.0342)	(0.0001)	(0.0245)
			1	Ours	0.0000	0.0320	0.0000	2.5308
					(0.0000)	(0.0839)	(0.0001)	(0.2418)
				ALM	13.3900	0.9280	0.6540	15.0569
					(0.6651)	(0.1223)	(0.0080)	(0.5758)
				GoDec	0.0000	0.0300	0.0001	2.5537
					(0.0000)	(0.0823)	(0.0003)	(0.2523)
	200	100	0.1	Ours	0.0000	0.9770	0.0000	0.2345
					(0.0000)	(0.0337)	(0.0000)	(0.0169)
				ALM	54.3100	1.0000	0.7034	4.9984
					(0.7745)	(0.0000)	(0.0022)	(0.0510)
				GoDec	0.0000	0.9755	0.0000	0.2330
					(0.0000)	(0.0344)	(0.0000)	(0.0160)
			1	Ours	0.0000	0.0075	0.0000	2.4469
					(0.0000)	(0.0206)	(0.0000)	(0.1387)
				ALM	54.2400	0.9456	0.7059	49.9838
					(0.7264)	(0.0456)	(0.0023)	(0.5095)
				GoDec	0.0000	0.0085	0.0000	2.4476
					(0.0000)	(0.0236)	(0.0000)	(0.1395)
$0.3p$	50	30	0.1	Ours	0.0000	0.9933	0.0002	0.2507
					(0.0000)	(0.0201)	(0.0003)	(0.0277)
				ALM	13.0000	1.0000	0.6472	1.5057
					(0.6195)	(0.0000)	(0.0079)	(0.0576)
				GoDec	0.0000	0.9953	0.0001	0.2489
					(0.0000)	(0.0171)	(0.0003)	(0.0271)
			1	Ours	0.0000	0.0373	0.0000	2.8870
					(0.0000)	(0.0624)	(0.0001)	(0.2410)
				ALM	13.37	0.9407	0.6531	15.0569
					(0.6301)	(0.0621)	(0.0080)	(0.5758)
				GoDec	0.0000	0.0327	0.0001	2.8983
					(0.0000)	(0.0653)	(0.0002)	(0.2504)
	200	100	0.1	Ours	0.0000	0.9867	0.0001	0.2495
					(0.0000)	(0.0164)	(0.0001)	(0.0198)
				ALM	54.3500	1.0000	0.7030	4.9984
					(0.6571)	(0.0000)	(0.0023)	(0.0510)
				GoDec	0.0000	0.9882	0.0000	0.2479
					(0.0000)	(0.0152)	(0.0001)	(0.0191)
			1	Ours	0.0000	0.0080	0.0000	2.8254
					(0.0000)	(0.0122)	(0.0000)	(0.1402)
				ALM	54.2200	0.9467	0.7054	49.9838
					(0.6289)	(0.0297)	(0.0022)	(0.5095)
				GoDec	0.0000	0.0075	0.0000	2.8237
					(0.0000)	(0.0135)	(0.0000)	(0.1409)

Table 3: Results for Simulation II . Algorithm 1 is used for computation.

are known. To identify the locations, we extract sparse ($\boldsymbol{\Theta}_1$) and low-rank ($\boldsymbol{\Theta}_2$) structures for the face images as described by the matrix decomposition into $\boldsymbol{\Theta}_1$ and $\boldsymbol{\Theta}_2$. For this purpose, $\boldsymbol{A}$ in (3) is set to be the identity matrix of size 171×171. Figures 2 and 3 display two decomposed structures for the AR face images by the proposed method with different sparse and rank constraint parameters in (3).

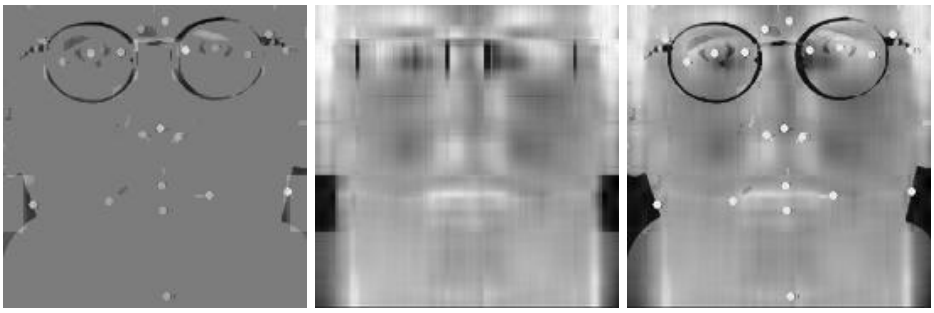


Figure 2: Extracted sparsity (first), low-rank (second) structures as well as the reconstructed image by the proposed method for AR face images; where the tuning parameters are set to $s_1 = 2500$, $s_2 = 5$.

Figure 3: Extracted sparsity (first), low-rank (second) structures as well as the reconstructed image by the proposed method for AR face images; where the tuning parameters are set to $s_1 = 2100$, $s_2 = 10$.

As indicated in Figures 2 and 3, the sparseness structure describes characteristics/detailed marks of the face, whereas the low-rank structure displays the rough outlook of the human face. This confirms our discussion regarding local and global features in the Introduction. Visually, both the first panels in Figures 2 and 3 preserve at least 60% markup points, especially the points around nose two sides of face and lip. In other words, the sparsity structure captures most of markup points. Similarly, the second panels retain the overall look of the face. Most interestingly, this decomposition tends to remove the glasses from the human face.

4.4 Greek Letters Image Reconstruction

Now consider a 26×31 black-white image of two Greek letters β and ϕ, where its noisy version is obtained by adding noise $N(0,1)$ after dividing the original matrix values by 100. The ratio of the maximum value of the image to the noise standard deviation is about 2.5. The images are displayed in Figure 4.

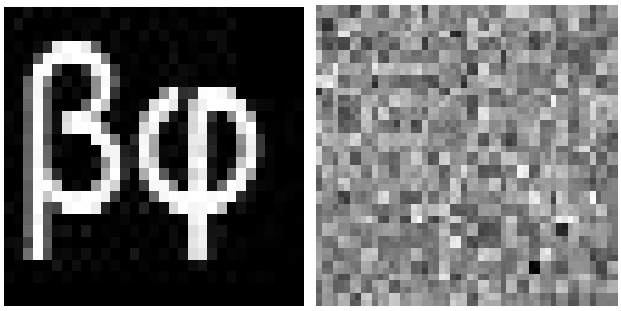


Figure 4: Original image (left) versus its noisy version (right).

Our goal is reconstruction of the original image from its noise version, with a focus on restoration of detailed structures of the letters. Towards this end, we apply the proposed method and contrast with its counterpart based on sparse pursuit alone and low-rank approximations. Specifically, let $\boldsymbol{A}$ to be the identity matrix of size 31×31 and $\boldsymbol{\Theta}$ be a 31×26 parameter matrix in (3). For each method, grid search is performed for tuning, with $s_1 = (10, 20, 30, 50)$, $1 \leq s_2 \leq \min(p, k) = 26$ and $\tau = (0.05, 0.1, 0.2)$. For each method, the 10-$fold$ cross-validation is employed. The reconstructed images are displayed in Figure 5.

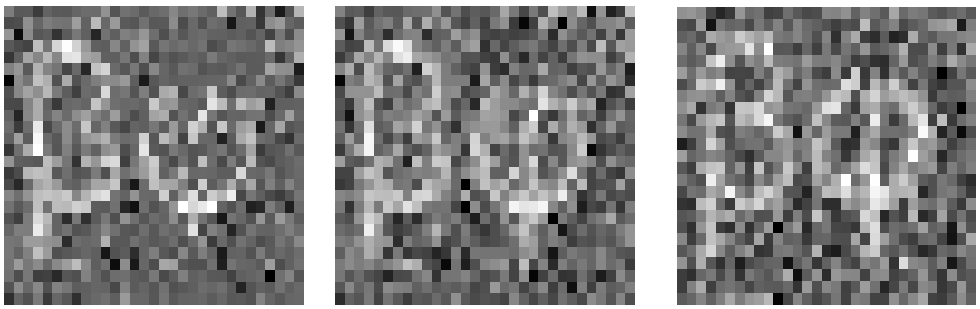


Figure 5: Reconstructed images based on sparsity alone (first), low-rank alone (second) and our method (third). Algorithm 2 is used for computation.

Visually, the first two reconstructed images by the low-rank method and the sparsity method give the rough shape of two letters, but the letters β and ϕ not distinguishable with blurred segments in places, especially the right middle of β and the top of ϕ. By comparison, the third reconstructed image by our method enables to reconstruct the complete shape of these two letters, and yield the best quality of reconstruction.

4.5 US Macroeconomic Time Series

This subsection examines multiple time series data described in (Stock & Watson, 2012). The data measures 143 US macroeconomic variables quarterly over a time span from February 1, 1959 to November 1, 2008. These variables are categorized into 13 groups and are summarized in Table 4.

Group	Description	Examples of series	# series
1	GDP component	GDP, consumption, investment	16
2	IP	IP, capacity utilization	14
3	Employment	Sectoral&total employment and hours	20
4	Unemployment rate	Unemployment rate, total and by duration	7
5	Housing	Housing starts, total and by region	6
6	Inventories	NAPM inventories, new orders	6
7	Prices	Price indexes, aggregate&disaggregate, commodity prices	37
8	Wages	Average hourly earning, unit labor cost	6
9	Interest rates	Treasuries, corporate, term spreads, public-private spreads	13
10	Money	M1, M2, business loans, consumer credit	7
11	Exchange rates	Average&selected trading partners	5
12	Stock prices	Various stock price indexes	5
13	Consumer expectations	Michigan consumer expectations	1

Table 4: Economic indicators collected for U.S. macroeconomic time series.

For data analysis, we consider time series starting from August 1, 1959 to November 1, 2008 due to incomplete initial observations. Our goal is one-step ahead forecasting, and contrast the proposed method with low-rank alone and sparsity alone in terms of forecasting accuracy. Using a multivariate autoregressive model, that is, $\boldsymbol{y}_t = \boldsymbol{y}_{t-1}^T \boldsymbol{\Theta} + \boldsymbol{\epsilon}_i$, we place it in the framework of (1), where $\boldsymbol{y}_t$ is a vector that records the values of various macroeconomic variables at time point t, and $\boldsymbol{\epsilon}_i$ follows normal distribution. In the presence of multiplicity and non-stationarity for economics data like this, we consider some transformations. For instance, log growth rates for quantity variables are differenced, nominal interest rates are differenced, as well as the logarithms of changes in rates of inflation for price series are differenced. See (Stock & Watson, 2012) for processing the data set. For this data set, $p = k = 143$ in (1) and the design matrix $\mathbf{A}$ is specified by the time series, which can written as $\boldsymbol{A} = (\boldsymbol{y}_{t_0}, \boldsymbol{y}_{t_0+1}, \ldots, \boldsymbol{y}_{t_0+d-1})^T$.

A one-step ahead K-fold cross validation (CV) criterion is used for tuning the time series (Arlot & Celisse, 2010). In particular, for design matrix $\boldsymbol{A}$, at each fold i, we use observations i to $n-K+i-1$ for training and the observation $n-K+i$ for tuning, where K is a pre-assigned integer and $K-1$ indicates the number of folds. Note that the values of p and k are close to the sample size n for this time series. We therefore choose $K \leq 20$ to maintain adequate training samples.

For tuning, the CV is optimized over a set of grids for $s_1 = (10, 20, 50, 100, 200)$, $1 \leq s_2 \leq \min(p, k)$ and $\tau = (0.02, 0.05, 0.1, 0.2)$. The results for $K = 11$ are reported in Table 5. The results for other K values are omitted due to similarity.

As suggested by Table 5, the proposed method outperforms its counterparts pursuing sparseness and low-rank alone. The amount of improvement over the low rank method and

	Ours	Low-rank alone	Sparsity alone
$K = 11$	301.22	348.02	3111.89

Table 5: Prediction errors of U.S. macroeconomic data for $K = 11$. Here "Low rank alone", "Sparsity alone" and "Ours" indicate our method for low rank pursuit only, for sparsity pursuit only and for simultaneous pursuit of low rank and sparsity. Algorithm 2 is used for computation.

the sparsity method is 15% and 933%, respectively. The Q-Q plots in Figure 6 indicate that the model assumption is adequate although some departure from normality has been detected. Overall, the proposed method performs reasonably well.

Acknowledgments

We would like to acknowledge support for this project from the National Science Foundation Grants IIS-0953662 and DMS-0906616, and National Institute of Health Grants R01 LM010730, 2R01GM081535 and 1R01HL105397. The authors thank the editor, the action editor and three referees for their helpful comments and suggestions.

Appendix

Proof of Lemma 1: Let $df(s,r) = s + (p+k-r)r$. By definition of the effective degrees of freedom, we obtain that

$$\text{Eff}(\boldsymbol{\Theta}) \leq \min(df(0, r(\boldsymbol{\Theta}^0)), df(\|\boldsymbol{\Theta}^0\|_0, 0)).$$

To prove uniqueness in terms of (s,r), suppose there exist $(\bar{s},\bar{r}) \neq (\bar{s}',\bar{r}')$ such that $df(\bar{s},\bar{r}) = df(\bar{s}',\bar{r}') = \min_{s,r} df(s,r)$. Without loss of generality, assume $\bar{r} = \bar{r}' - n_0 < \bar{r}'$, where $n_0 > 0$ is a positive integer. If $n_0 \leq \min(p,k) - \bar{r}$ and $\bar{r} < \min(p,k)$, then $\bar{s} + (p+k-\bar{r})\bar{r} = \bar{s}' + (p+k-\bar{r}')\bar{r}'$ implies that $\bar{s} = \bar{s}' + n_0(p+k-2\bar{r}-n_0) \geq n_0(p+k-2\bar{r}-n_0) > p+k-2\bar{r}-1$, which contradicts with the assumption that $s < p+k-2r-1$. Otherwise, if $\bar{r} = \min(p,k)$, $\bar{s}$ must be zero. This completes the proof. ■

Proof of Lemma 2: Let $x_i = v_i$ for $i \notin K$. Then the problem reduces to the standard l_1 ball problem.

$$\underset{\sum_{i\in K}|x_i|\leq z}{\text{argmin}} \frac{1}{2}\sum_{i\in K}(x_i - v_i)^2.$$

The results follows by the proof of Theorem 1 of (Liu & Ye, 2009). ■

Proof of Lemma 3: It suffices to derive the basic step of ISTA in (Amit et al., 2007) for (13). Consider the following quadratic approximation of problem (13) at a given point $\boldsymbol{y}$:

$$\min_{\boldsymbol{x}\in\mathbb{R}^n:\sum_{i\in K}|x_i|\leq z} Q_L(\boldsymbol{x},\boldsymbol{y}) = \|\boldsymbol{A}\boldsymbol{y} - \boldsymbol{b}\|_2^2 + \langle \boldsymbol{x}-\boldsymbol{y}, \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{y}-\boldsymbol{b})\rangle + \frac{L}{2}\|\boldsymbol{x}-\boldsymbol{y}\|_2^2, \qquad (20)$$

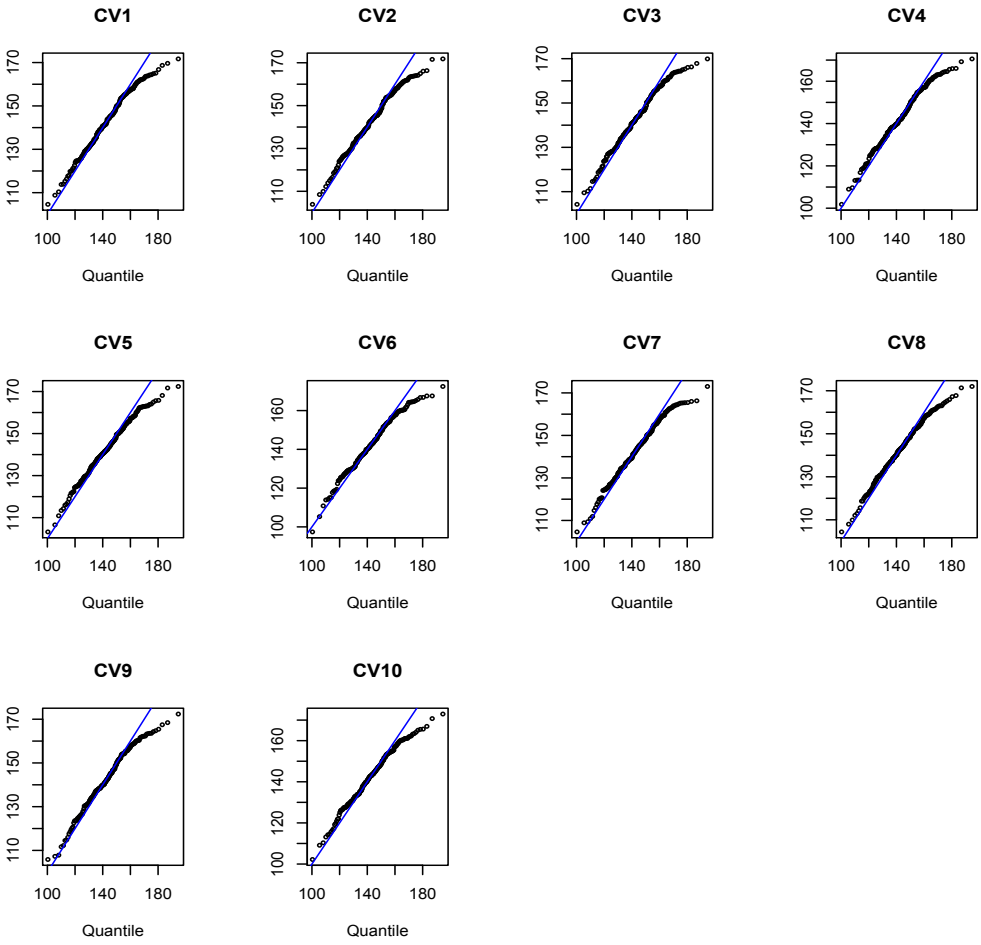


Figure 6: Q-Q plots for each-fold in U.S. macroeconomic time series data example, where points on a straight line indicates non-departure from normality.

where L is a Lipschitz constant of the function $\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b})$ with respect to $\boldsymbol{x}$. Solving (20) is equivalent to that of

$$\min_{\boldsymbol{x}\in\mathbb{R}^n:\sum_{i\in K}|x_i|\le z} \|\boldsymbol{x}-\big(\boldsymbol{y}-\frac{1}{L}\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{y}-\boldsymbol{b})\big)\|_2^2.$$

By Lemma 2, the solution is $\mathcal{T}_{K,z}\big(\boldsymbol{y}-\frac{1}{L}\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{y}-\boldsymbol{b})\big)$. The basic step of ISTA thus can be written as $\boldsymbol{x}^{(t)}=\mathcal{T}_{K,z}\big(\boldsymbol{x}^{(t-1)}-\frac{1}{L}\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{x}^{(t-1)}-\boldsymbol{b})\big)$. Then, Lemma 3 follows by taking L to be $\lambda_{\max}(\boldsymbol{A}^T\boldsymbol{A})$, where $\lambda_{\max}(\cdot)$ denotes the largest singular value. ■

Proof of Lemma 4: By (6) and (7), for any integer $m\ge 1$,

$$f(\hat{\boldsymbol{\Theta}}_1^{(m)},\hat{\boldsymbol{\Theta}}_2^{(m)})\ge f(\hat{\boldsymbol{\Theta}}_1^{(m)},\hat{\boldsymbol{\Theta}}_2^{(m+1)})\ge f(\hat{\boldsymbol{\Theta}}_1^{(m+1)},\hat{\boldsymbol{\Theta}}_2^{(m+1)}).$$

Meanwhile, it follows from (6) that

$$\begin{aligned} f(\hat{\boldsymbol{\Theta}}_1^{(m)},\hat{\boldsymbol{\Theta}}_2^{(m)}) &= \|\boldsymbol{Z}-\hat{\boldsymbol{\Theta}}_2^{(m)}\|_F^2-\|\hat{\boldsymbol{\Theta}}_1^{(m)}\|_F^2\\ &\ge \|\boldsymbol{Z}-\hat{\boldsymbol{\Theta}}_2^{(m+1)}-\hat{\boldsymbol{\Theta}}_1^{(m)}\|_F^2\\ &\ge \|\boldsymbol{Z}-\hat{\boldsymbol{\Theta}}_2^{(m+1)}\|_F^2-\|\hat{\boldsymbol{\Theta}}_1^{(m)}\|_F^2. \end{aligned}$$

Therefore $\|\boldsymbol{Z}-\hat{\boldsymbol{\Theta}}_2^{(m)}\|_F^2$ is lower bounded and decreasing in m. Moreover, by the monotone properties of $f(\hat{\boldsymbol{\Theta}}_1^{(m)},\hat{\boldsymbol{\Theta}}_2^{(m)})$, $\|\hat{\boldsymbol{\Theta}}_1^{(m)}\|_F^2$ converges as $m\to\infty$. Then there exists a subsequence $\{m_k\}$ such that $(\hat{\boldsymbol{\Theta}}_1^{(m_k)},\hat{\boldsymbol{\Theta}}_2^{(m_k)})\to(\hat{\boldsymbol{\Theta}}_1^{(m^*)},\hat{\boldsymbol{\Theta}}_2^{(m^*)})$.

Let $R_{ij}(\boldsymbol{\Theta}_1,\boldsymbol{\Theta}_2)\in\operatorname{argmin}_{\theta_{ij}\in\boldsymbol{\Theta}_1 \text{ or } \theta_{ij}\in\boldsymbol{\Theta}_2} f((\boldsymbol{\Theta}_1,\boldsymbol{\Theta}_2)\setminus\theta_{ij})$. Let the cost function for θ_{ij} to be $f_m(\theta_{ij})=f((\hat{\boldsymbol{\Theta}}_1^{(m)},\hat{\boldsymbol{\Theta}}_2^{(m)})\setminus\theta_{ij})$, where other components of $(\hat{\boldsymbol{\Theta}}_1^{(m)},\hat{\boldsymbol{\Theta}}_2^{(m)})$ are held fixed. Then

$$\begin{aligned} f_{m_k}(R_{ij}(\hat{\boldsymbol{\Theta}}_1^{(m^*)},\hat{\boldsymbol{\Theta}}_2^{(m^*)})) &\ge f_{m_k}(R_{ij}(\hat{\boldsymbol{\Theta}}_1^{(m_k)},\hat{\boldsymbol{\Theta}}_2^{(m_k)}))\\ &\ge \min\Big(f((\hat{\boldsymbol{\Theta}}_1^{(m_k)},\hat{\boldsymbol{\Theta}}_2^{(m_k+1)})),f((\hat{\boldsymbol{\Theta}}_1^{(m_k)},\hat{\boldsymbol{\Theta}}_2^{(m_k)}))\Big)\\ &\ge f((\hat{\boldsymbol{\Theta}}_1^{(m_k+1)},\hat{\boldsymbol{\Theta}}_2^{(m_k+1)})). \end{aligned}$$

As $m\to\infty$, by continuity of $f(\cdot)$, $f_{(m^*)}(R_{ij}(\hat{\boldsymbol{\Theta}}_1^{(m^*)},\hat{\boldsymbol{\Theta}}_2^{(m^*)}))\ge f(\hat{\boldsymbol{\Theta}}_1^{(m^*)},\hat{\boldsymbol{\Theta}}_2^{(m^*)}))$, where the equality holds by the definition of R_{ij}. Hence, for each $\theta_{ij}\in\boldsymbol{\Theta}_l$; $l=1,\ 2$, $\hat{\theta}_{ij}^{(m^*)}=R_{ij}(\hat{\boldsymbol{\Theta}}_1^{(m^*)},\hat{\boldsymbol{\Theta}}_2^{(m^*)})$ is the optimal componentwise solution. The results of Lemma 4 then follow. ■

Proof of Lemma 5: First we prove that $\hat{\boldsymbol{\Theta}}_1^{(m)}$ satisfies

$$\sum_{\theta_{ij}\in\boldsymbol{\Theta}_1}\left(\frac{|\hat{\theta}_{ij}^{(m)}|}{\tau}I(|\hat{\theta}_{ij}^{(m)}|\le\tau)+I(|\hat{\theta}_{ij}^{(m)}|>\tau)\right)\le s_1. \tag{21}$$

Toward this end, we rewrite the left side of (21) as

$$\begin{aligned}
&\sum_{\theta_{ij}\in\mathbf{\Theta}_1}\left(\frac{|\hat{\theta}_{ij}^{(m)}|}{\tau}I(|\hat{\theta}_{ij}^{(m-1)}|\le\tau)+I(|\hat{\theta}_{ij}^{(m-1)}|>\tau)\right)\\
&+\sum_{\theta_{ij}\in\mathbf{\Theta}_1}\left(\frac{|\hat{\theta}_{ij}^{(m)}|}{\tau}I(|\hat{\theta}_{ij}^{(m)}|\le\tau)+I(|\hat{\theta}_{ij}^{(m)}|>\tau)-\frac{|\hat{\theta}_{ij}^{(m)}|}{\tau}I(|\hat{\theta}_{ij}^{(m-1)}|\le\tau)-I(|\hat{\theta}_{ij}^{(m-1)}|>\tau)\right)\\
&=\sum_{\theta_{ij}\in\mathbf{\Theta}_1}\left(\frac{|\hat{\theta}_{ij}^{(m)}|}{\tau}I(|\hat{\theta}_{ij}^{(m-1)}|\le\tau)+I(|\hat{\theta}_{ij}^{(m-1)}|>\tau)\right)+I_m, \qquad (22)
\end{aligned}$$

where $I_m=\sum_{\theta_{ij}\in\mathbf{\Theta}_1}\frac{|\hat{\theta}_{ij}^{(m)}|-\tau}{\tau}\Big(I(|\hat{\theta}_{ij}^{(m)}|\le\tau)-I(|\hat{\theta}_{ij}^{(m-1)}|\le\tau)\Big)$. Note that it follows from the DC construction that

$$\sum_{\theta_{ij}\in\mathbf{\Theta}_1}\left(\frac{|\hat{\theta}_{ij}^{(m)}|}{\tau}I(|\hat{\theta}_{ij}^{(m-1)}|\le\tau)+I(|\hat{\theta}_{ij}^{(m-1)}|>\tau)\right)\le s_1.$$

Thus, to establish (21), we only need to prove $I_m\le 0$. Rewrite I as

$$I_m=\begin{cases}0 & \text{if } \min(|\hat{\theta}_{ij}^{(m)}|,|\hat{\theta}_{ij}^{(m-1)}|)>\tau \text{ or } \max(|\hat{\theta}_{ij}^{(m)}|,|\hat{\theta}_{ij}^{(m-1)}|)\le\tau,\\ \sum_{\theta_{ij}\in\mathbf{\Theta}_1}(\frac{|\hat{\theta}_{ij}^{(m)}|}{\tau}-1) & \text{if } |\hat{\theta}_{ij}^{(m)}|\le\tau \text{ and } |\hat{\theta}_{ij}^{(m-1)}|>\tau,\\ -\sum_{\theta_{ij}\in\mathbf{\Theta}_1}(\frac{|\hat{\theta}_{ij}^{(m)}|}{\tau}-1) & \text{if } |\hat{\theta}_{ij}^{(m)}|>\tau \text{ and } |\hat{\theta}_{ij}^{(m-1)}|\le\tau,\end{cases}$$

implying that $I_m\le 0$. Then, (21) follows.

For stationarity, note that it follows from (21) that

$$f(\hat{\mathbf{\Theta}}_1^{(m-1)},\hat{\mathbf{\Theta}}_2^{(m-1)})\ge f(\hat{\mathbf{\Theta}}_1^{(m-1)},\hat{\mathbf{\Theta}}_2)\ge f(\hat{\mathbf{\Theta}}_1^{(m)},\hat{\mathbf{\Theta}}_2^{(m)}),$$

where $\hat{\mathbf{\Theta}}_2$ is defined in Step 2 of **Algorithm 2**.

Suppose that termination index m^* is infinite. Then we will prove that $\hat{\mathbf{\Theta}}_1^{(m)}\to\hat{\mathbf{\Theta}}_1^{(m^*)}$ as $m\to m^*=\infty$. When $m^*=\infty$, $\hat{\mathbf{\Theta}}_1^{(m)}$ must be updated infinitely because $\hat{\mathbf{\Theta}}_2^{(m)}$ is analytically solved. First consider, at step m, $\mathbf{\Theta}_1$ is updated whereas $\mathbf{\Theta}_2=\hat{\mathbf{\Theta}}_2^{(m)}$. Denote by $\Lambda(\mathbf{\Theta}_1,\mathbf{\Theta}_2,\lambda^*)$ the dual problem of (12), where λ^* is the optimal Lagrange multiplier and $\mathbf{\Theta}_2=\hat{\mathbf{\Theta}}_2^m$. Then

$$\begin{aligned}
f(\hat{\mathbf{\Theta}}_1^{(m)},\hat{\mathbf{\Theta}}_2^{(m)})-f(\hat{\mathbf{\Theta}}_1^{(m+1)},\hat{\mathbf{\Theta}}_2^{(m+1)})&=\Lambda(\hat{\mathbf{\Theta}}_1^{(m)},\hat{\mathbf{\Theta}}_2^{(m)},\lambda^*)-\Lambda(\hat{\mathbf{\Theta}}_1^{(m+1)},\hat{\mathbf{\Theta}}_2^{(m)},\lambda^*)\\
&\quad-\lambda^*\sum_{\theta_{ij}\in\mathbf{\Theta}_1}\left(\frac{|\hat{\theta}_{ij}^{(m)}|}{\tau}I(|\hat{\theta}_{ij}^{(m)}|\le\tau)+I(|\hat{\theta}_{ij}^{(m)}|>\tau)-s_1\right)
\end{aligned}$$

The equality holds because $\hat{\mathbf{\Theta}}_1^{(m+1)}$ is the global minimizer of a convex problem (12), attaining at constraint boundaries, i.e $\sum_{\theta_{ij}\in\mathbf{\Theta}_1}\left(\frac{|\hat{\theta}_{ij}^{(m+1)}|}{\tau}I(|\hat{\theta}_{ij}^{(m)}|\le\tau)+I(|\hat{\theta}_{ij}^{(m)}|>\tau)-s_1\right)=0$.

An application of the Taylor expansion to $\Lambda(\boldsymbol{\Theta}_1, \hat{\boldsymbol{\Theta}}_2^{(m)}, \lambda^*)$ at $\boldsymbol{\Theta}_1 = \hat{\boldsymbol{\Theta}}_1^{(m+1)}$ yields that

$$\begin{aligned}
&f(\hat{\boldsymbol{\Theta}}_1^{(m)}, \hat{\boldsymbol{\Theta}}_2^{(m)}) - f(\hat{\boldsymbol{\Theta}}_1^{(m+1)}, \hat{\boldsymbol{\Theta}}_2^{(m+1)}) \\
&= \langle \frac{\partial \Lambda}{\partial \boldsymbol{\Theta}_1}(\hat{\boldsymbol{\Theta}}_1^{(m+1)}, \hat{\boldsymbol{\Theta}}_2^{(m)}, \lambda^*), \hat{\boldsymbol{\Theta}}_1^{(m)} - \hat{\boldsymbol{\Theta}}_1^{(m+1)} \rangle + \frac{1}{2}\langle \boldsymbol{A}(\hat{\boldsymbol{\Theta}}_1^{(m)} - \hat{\boldsymbol{\Theta}}_1^{(m+1)}), \boldsymbol{A}(\hat{\boldsymbol{\Theta}}_1^{(m)} - \hat{\boldsymbol{\Theta}}_1^{(m+1)}) \rangle \\
&\quad - \lambda^* \sum_{\theta_{ij} \in \boldsymbol{\Theta}_1} \left(\frac{|\hat{\theta}_{ij}^{(m)}|}{\tau} I(|\hat{\theta}_{ij}^{(m)}| \le \tau) + I(|\hat{\theta}_{ij}^{(m)}| > \tau) - s_1 \right),
\end{aligned}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product. The first term in the right side of the equality is zero, because $\hat{\boldsymbol{\Theta}}_1^{(m+1)}$ is the global minimizer and the third term is no less than zero by (21). Thus,

$$\begin{aligned}
f(\hat{\boldsymbol{\Theta}}_1^{(m)}, \hat{\boldsymbol{\Theta}}_2^{(m)}) - f(\hat{\boldsymbol{\Theta}}_1^{(m+1)}, \hat{\boldsymbol{\Theta}}_2^{(m+1)}) &\ge \frac{1}{2}\langle \boldsymbol{A}(\hat{\boldsymbol{\Theta}}_1^{(m)} - \hat{\boldsymbol{\Theta}}_1^{(m+1)}), \boldsymbol{A}(\hat{\boldsymbol{\Theta}}_1^{(m)} - \hat{\boldsymbol{\Theta}}_1^{(m+1)}) \rangle \\
&\ge \frac{\lambda_{\min}(\boldsymbol{A}^T\boldsymbol{A})}{2} \|\hat{\boldsymbol{\Theta}}_1^{(m)} - \hat{\boldsymbol{\Theta}}_1^{(m+1)}\|_F^2, \qquad (23)
\end{aligned}$$

where $\lambda_{\min(\cdot)}$ is the smallest eigenvalue of a matrix. Therefore $f(\hat{\boldsymbol{\Theta}}_1^{(m)}, \hat{\boldsymbol{\Theta}}_2^{(m)})$ is lower bounded and decreasing in m, implying $f(\hat{\boldsymbol{\Theta}}_1^{(m)}, \hat{\boldsymbol{\Theta}}_2^{(m)})$ converges to some limit f^* as $m \to \infty$. By (23), convergence of $\hat{\boldsymbol{\Theta}}_1^{(m)} \to \hat{\boldsymbol{\Theta}}_1^{(m^*)}$ is established. Next consider the case in which $\boldsymbol{\Theta}_2$ is only updated finitely, say before step m_0, using the same notation with proof of Lemma 4, then for any $m > m_0$

$$f_m(R_{ij}(\hat{\boldsymbol{\Theta}}_1^{(m^*)}, \hat{\boldsymbol{\Theta}}_2^{(m^*)})) \ge f_m(R_{ij}(\hat{\boldsymbol{\Theta}}_1^{(m)}, \hat{\boldsymbol{\Theta}}_2^{(m)})) = f((\hat{\boldsymbol{\Theta}}_1^{(m+1)}, \hat{\boldsymbol{\Theta}}_2^{(m+1)})).$$

The second equality holds because the MBI is employed. As $m \to m^*$, by continuity of function f, $f_{(m^*)}(R_{ij}(\hat{\boldsymbol{\Theta}}_1^{(m^*)}, \hat{\boldsymbol{\Theta}}_2^{(m^*)})) \ge f(\hat{\boldsymbol{\Theta}}_1^{(m^*)}, \hat{\boldsymbol{\Theta}}_2^{(m^*)}))$, where the equality holds by the definition of R_{ij}. Finally, we consider the case in which $\boldsymbol{\Theta}_2$ is updated infinitely. Then there is a subsequence $\{m_k\}$ such that $\hat{\boldsymbol{\Theta}}_2^{(m_k)} \to \hat{\boldsymbol{\Theta}}_2^{(m^*)}$. Similarly, $f_{m^*}(R_{ij}(\hat{\boldsymbol{\Theta}}_1^{(m^*)}, \hat{\boldsymbol{\Theta}}_2^{(m^*)})) = f(\hat{\boldsymbol{\Theta}}_1^{(m^*)}, \hat{\boldsymbol{\Theta}}_2^{(m^*)}))$. Hence, for each $\theta_{ij} \in \boldsymbol{\Theta}_l$, $l = 1,\ 2$, $\hat{\theta}_{ij}^{(m^*)} = R_{ij}(\hat{\boldsymbol{\Theta}}_1^{(m^*)}, \hat{\boldsymbol{\Theta}}_2^{(m^*)})$ is the optimal componentwise solution. The results of Lemma 5 then follow. ■

Let $\mathcal{B}_{S,r} = \{\boldsymbol{\Theta} = \boldsymbol{\Theta}_1^S + \boldsymbol{\Theta}_2 : r(\boldsymbol{\Theta}_2) = r\} \cap \Lambda$, a sub-parameter space with known sparsity structure S and rank r. Denote $H(\cdot, \Lambda)$ and $H^B(\cdot, \Lambda)$ to be the L_∞ entropy and bracketing Hellinger metric entropy for set Λ, respectively. The next two technical lemmas concern the size of the parameter space.

Lemma 9 *Suppose that Assumptions A is met.*

$$H^B(t, \mathcal{B}_{S,r}) \le |S| \log(2Ml_1/t) + (p+k)r \log(2Ml_2^3/t),$$

where l_1, l_2 are constant and $M > 1$ is defined in Assumption A.

Lemma 10 *Suppose that Assumptions A is satisfied. If $s_1 = s_1^0$, $s_2 = s_2^0$, then*

$$\begin{aligned}
H^B(t, \Lambda) \le& 2(p+k)s_2^0 \log(2Ml_2^3\epsilon/t) + s_1^0 \log((1+2Ml_1)/t) \\
&+ 2s_1^0 \log\left(e\frac{(p+k-r(\boldsymbol{\Theta}^0))r(\boldsymbol{\Theta}^0)}{s_1^0} \right).
\end{aligned}$$

Proof of Lemmas 6 and 7: For Lemma 6, note that $\Lambda = \cup_{|S|\le s_1^0} \cup_{r\le s_2^0} \mathcal{B}_{S,r}$. It suffices to calculate the entropy for each $\mathcal{B}_{S,r}$.

Let $\Lambda_2 = \{(\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2) : \boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2 \text{ satisfy conditions defined in } \Lambda\}$. For $\boldsymbol{\Theta} = \boldsymbol{\Theta}_1 + \boldsymbol{\Theta}_2$ and $(\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2) \in \Lambda_2$, define $\mathcal{B}_\delta(\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2) = \{(\boldsymbol{\Theta}_1', \boldsymbol{\Theta}_2') \in \Lambda_2 : \|\boldsymbol{\Theta}_1 - \boldsymbol{\Theta}_1'\|_{\max} + \|\boldsymbol{\Theta}_2 - \boldsymbol{\Theta}_2'\|_{\max} \le \delta\}$ to be the neighborhood of $(\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2)$. For any $\boldsymbol{\Theta}' = \boldsymbol{\Theta}_1' + \boldsymbol{\Theta}_2'$ with $(\boldsymbol{\Theta}_1', \boldsymbol{\Theta}_2') \in \mathcal{B}_\delta(\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2)$, by Assumption A,

$$\int \sup_{\mathcal{B}_\delta(\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2)} (g^{1/2}(\boldsymbol{\Theta}, y) - g^{1/2}(\boldsymbol{\Theta}', y))^2 d\nu(y) \le M^2\delta^2.$$

Combined the above with Lemma 2.1 of (Ossiander, 1987), we have

$$H^B(t, \mathcal{B}_{S,r}) \le H(M^{-1}t, \mathcal{B}_{S,r}). \tag{24}$$

Since $\|\boldsymbol{\Theta}_1\|_{\max}$ is bounded by l_1, by constructing a $2t$-net on $\mathcal{B}_{S,r}$ through the outer product of the t-nets on $\boldsymbol{\Theta}_1^S$ and $\boldsymbol{\Theta}_2$ defined in the parameter space Λ, we can show that

$$H(M^{-1}t, \mathcal{B}_{S,r}) \le |S| \log(2Ml_1/t) + H_r(M^{-1}t) \tag{25}$$

where $|S|$ is the number of nonzeros in $\boldsymbol{\Theta}_1$ and $H_r(M^{-1}t)$ is the entropy for $\boldsymbol{\Theta}_2$ with rank r. Let $\boldsymbol{C}$ be a basis of column of $\boldsymbol{\Theta}_2$, then there exists an $k \times r$ matrix $\boldsymbol{F}$ such that $\boldsymbol{\Theta}_2 = \boldsymbol{C}\boldsymbol{F}$. Hence

$$\|\boldsymbol{\Theta}_2 - \boldsymbol{\Theta}_2'\|_{\max} = \|\boldsymbol{C}\boldsymbol{F} - \boldsymbol{C}'\boldsymbol{F}'\|_{\max} \le \|\boldsymbol{C}\|_\infty \|\boldsymbol{F} - \boldsymbol{F}'\|_{\max} + \|\boldsymbol{F}^T\|_\infty \|\boldsymbol{C} - \boldsymbol{C}'\|_{\max}.$$

where $\|\boldsymbol{\Theta}_{p\times k}\|_\infty = \max_{1\le i\le p} \sum_{i=1}^k |\theta_{ij}|$ is the L_∞ matrix-norm and $\|\boldsymbol{\Theta}\|_{\max} = \max_{\theta_{ij}\in\boldsymbol{\Theta}} |\theta_{ij}|$ is the max norm. Note that $\|\boldsymbol{C}\|_\infty$ and $\|\boldsymbol{F}^T\|_\infty$ are bounded by l_2. This yields

$$H_r(M^{-1}t) \le (p+k) r \log \frac{2l_2^3 M}{t}.$$

This, together with (24) and (25), implies Lemma 6.

For Lemma 7, note that

$$\begin{aligned}
&\exp(H^B(t,\Lambda)) \le \exp(H(M^{-1}t, \Lambda)) \\
&= \sum_{r=0}^{s_2^0} \sum_{|S|=0}^{s_1^0} \sum_{i=0}^{|S|} \binom{s_1^0}{i} \binom{(p+k-r(\boldsymbol{\Theta}^0))r(\boldsymbol{\Theta}^0) - s_1^0}{|S|-i} \exp(H(M^{-1}t, \mathcal{B}_{S,r})) \\
&\le \binom{(p+k-r(\boldsymbol{\Theta}^0))r(\boldsymbol{\Theta}^0)}{s_1^0} \left(\sum_{|S|=0}^{s_1^0} \binom{s_1^0}{|S|} (2Ml_1/t)^{|S|} \right) \left(\sum_{r=0}^{s_2^0} (2Ml_2^3/t)^{(p+k)r} \right) \\
&\equiv \binom{(p+k-r(\boldsymbol{\Theta}^0))r(\boldsymbol{\Theta}^0)}{s_1^0} \times I \times II.
\end{aligned}$$

Note that $\sum_{k=0}^n \binom{n}{k} a^k b^{n-k} = (a+b)^n$. Then $I = (1+\frac{2Ml_1}{t})^{s_1^0}$ and $II \le (s_2^0+1) \left(\frac{2Ml_2^3\epsilon}{t}\right)^{(p+k)s_2^0}$. Thus,

$$\begin{aligned}
H^B(t,\Lambda) &\le \log \binom{(p+k-r(\boldsymbol{\Theta}^0))r(\boldsymbol{\Theta}^0)}{s_1^0} + \log(s_2^0+1) + s_1^0 \log(1+\frac{2Ml_1}{t}) + (p+k)s_2^0 \log(\frac{2Ml_2^3}{t}) \\
&\le 2(p+k)s_2^0 \log(2Ml_2^3/t) + s_1^0 \log(\frac{1+2Ml_1}{t}) + 2s_1^0 \log\left(e\frac{(p+k-r(\boldsymbol{\Theta}^0))r(\boldsymbol{\Theta}^0)}{s_1^0}\right),
\end{aligned}$$

where e is the natural number and $0 < t < 1$. The last inequality follows Theorem 2.6 of (Stanica & Montgomery, 2001) that $\binom{b}{a} \leq \frac{b^{b+1/2}}{\sqrt{2\pi}a^{a+1/2}(b-a)^{b-a+1/2}} \leq \exp((a+1/2)\log(b/a) + a) \leq \exp(2a\log(b/a)+a)$ for any integer $0 < a < b$. This completes the proof. ■

Proof of Theorem 1: We apply a large deviation inequality in Theorem 2 of (Wong & Shen, 1995). To this end, we verify (1.2) there. By Lemma 7,

$$\int_{\epsilon^2/2^8}^{2^{1/2}\epsilon} \left(H^B(t/c_4, \Lambda)\right)^{1/2} dt \leq \int_{\epsilon^2/2^8}^{2^{1/2}\epsilon} \sqrt{2(p+k)s_2^0 \log(2Ml_2^3c_4/t) + s_1^0 \log((1+2Ml_1)c_4/t)}dt$$
$$+ \int_{\epsilon^2/2^8}^{2^{1/2}\epsilon} \sqrt{2s_1^0 \log\left(e\frac{(p+k-r(\mathbf{\Theta}^0))r(\mathbf{\Theta}^0)}{s_1^0}\right)}dt \equiv I_1 + I_2,$$

for some constant $c_4 > 0$, say $c_4 = 10$. Then, for ϵ small,

$$\begin{aligned} I_1 &\leq \sqrt{2}\epsilon\sqrt{2(p+k)s_2^0 \log(2^9Ml_2^3c_4/\epsilon^2) + s_1^0 \log((1+2Ml_1)2^8c_4/\epsilon^2)} \\ &\leq 2\epsilon\sqrt{(p+k)s_2^0 + s_1^0}\sqrt{\log(2^9Mc_4(l_2^3+l_1)) + 2\log\frac{1}{\epsilon}} \\ &\leq 2\sqrt{2}\epsilon\sqrt{\log(2^9Mc_4(l_2^3+l_1))}\sqrt{(p+k)s_2^0 + s_1^0} \cdot \sqrt{\log\frac{1}{\epsilon}}. \end{aligned}$$

Similarly,

$$I_2 \leq 2\epsilon\sqrt{s_1^0 \log\left(e\frac{(p+k-r(\mathbf{\Theta}^0))r(\mathbf{\Theta}^0)}{s_1^0}\right)}.$$

Let $\epsilon_{n,p,k} = \frac{C_{p,k}}{\sqrt{n}}\log(\frac{C_{p,k}}{\sqrt{n}})$ where

$$C_{p,k} = 2\sqrt{2}c_5^{-1}\sqrt{\log(2^9Mc_4(l_2^3+l_1))}\sqrt{(p+k)s_2^0 + s_1^0} + 2c_5^{-1}\sqrt{s_1^0 \log\left(e\frac{(p+k-r(\mathbf{\Theta}^0))r(\mathbf{\Theta}^0)}{s_1^0}\right)}.$$

Then, for any $\epsilon \geq \epsilon_{n,p,k}$ and $c_5 = \frac{512}{(2/3)^{5/12}}$

$$\int_{\epsilon^2/2^8}^{2^{1/2}\epsilon} \left(H^B(t/c_4, \Lambda)\right)^{1/2} dt \leq c_5^{-1}\sqrt{n}\epsilon^2.$$

By Theorem 2 of (Wong & Shen, 1995), $P\left(h(\hat{\mathbf{\Theta}}^{L_0}, \mathbf{\Theta}^0) \geq \epsilon\right) \leq 5\exp(-c_1 n\epsilon^2)$, which yields $Eh^2(\hat{\mathbf{\Theta}}^{L_0}, \mathbf{\Theta}^0) = O(\epsilon_{n,p,k}^2)$ by using the fact that $h(\hat{\mathbf{\Theta}}^{L_0}, \mathbf{\Theta}^0) \leq 1$.

Consider a special situation when $\log(r(\mathbf{\Theta}^0)) \leq ds_2^0$ for some constant $d > 0$ that is independent of p, k. Note that $s_1^0 < p + k - s_2^0$ and $p + k - r(\mathbf{\Theta}^0) \leq p + k - s_2^0$. Then

$$\begin{aligned} s_1^0 \log\left(e\frac{(p+k-r(\mathbf{\Theta}^0))r(\mathbf{\Theta}^0)}{s_1^0}\right) &\leq (p+k-s_2^0)\log\left(e\frac{(p+k-r(\mathbf{\Theta}^0))r(\mathbf{\Theta}^0)}{p+k-s_2^0}\right) \\ &\leq (p+k-s_2^0)\log(er(\mathbf{\Theta}^0)) \leq 2d(p+k-s_2^0)s_2^0. \end{aligned}$$

Thus, $I_1 + I_2$ is upper bounded by

$$2\sqrt{2}\epsilon\left(\sqrt{\log(2^9 M c_4(l_2^3+l_1))}+\sqrt{d}\right)\sqrt{(p+k)s_2^0+s_1^0}\cdot\sqrt{\log\frac{1}{\epsilon}}.$$

Let $c_3 = 2\sqrt{2}c_5^{-1}\left(\sqrt{\log(2^9 c_4(l_2^3+l_1))}+\sqrt{d}\right)\sqrt{(p+k)s_2^0+s_1^0}$. The result then follows. This completes the proof. ■

Proof of Corollary 1: If $\mathbf{\Theta}^0$ is sparse and $\|\mathbf{\Theta}^0\|_0 \le p+k-2$, then by the definition of effective degrees of freedom $s_0 = s_1^0 + (p+k-s_2^0)s_2^0 \le \|\mathbf{\Theta}^0\|_0$. This implies that

$$\begin{aligned} C_{p,k} &= O\left(\sqrt{\|\mathbf{\Theta}^0\|_0}\right) + O\left(\sqrt{\|\mathbf{\Theta}^0\|_0 \log\left(p+k-r(\mathbf{\Theta}^0))r(\mathbf{\Theta}^0)/\|\mathbf{\Theta}^0\|_0\right)}\right) \\ &= O\left(\sqrt{\|\mathbf{\Theta}^0\|_0 \log\left(p+k-r(\mathbf{\Theta}^0))r(\mathbf{\Theta}^0)/\|\mathbf{\Theta}^0\|_0\right)}\right). \end{aligned}$$

The second inequality is because of nondecreasingness of $\sqrt{x}$ and $\sqrt{x\log(a/x)}$ in x for $x \le a/e$.

If $\mathbf{\Theta}^0$ is low-rank, we have

$$C_{p,k} = O\left(\sqrt{(p+k-r(\mathbf{\Theta}^0))r(\mathbf{\Theta}^0)}+\sqrt{s_1^0\log\left((p+k-r(\mathbf{\Theta}^0))r(\mathbf{\Theta}^0)/s_1^0\right)}\right).$$

Note that $s_1^0\log\left((p+k-r(\mathbf{\Theta}^0))r(\mathbf{\Theta}^0)/s_1^0\right) \le \log\left((p+k-r(\mathbf{\Theta}^0))r(\mathbf{\Theta}^0)/e\right)$. The result follows.

If $\mathbf{\Theta}^0$ is dense and of full rank, then $(p+k-r(\mathbf{\Theta}^0))r(\mathbf{\Theta}^0)$ is of order $O(pk)$. Hence $C_{p,k}$ can be written as $O\left(\sqrt{(p+k-s_2^0)s_2^0}+\sqrt{s_1^0\log(\frac{pk}{s_1^0})}\right)$. This completes the proof. ■

Proof of Corollary 2: It suffices to show the Assumption A is met. Let $f(\boldsymbol{\mu}_i, \boldsymbol{y}) = \frac{1}{(\sqrt{2\pi}\sigma)^k}\exp\left(-\frac{1}{2\sigma^2}(\boldsymbol{y}-\boldsymbol{\mu}_i)^T(\boldsymbol{y}-\boldsymbol{\mu}_i)\right)$ for $i = 1, 2$. $\boldsymbol{\mu}_1 = \boldsymbol{a}^T\mathbf{\Theta}$ and $\boldsymbol{\mu}_2 = \boldsymbol{a}^T\mathbf{\Theta}'$. Then

$$\begin{aligned} &\int \sup_{\|\mathbf{\Theta}-\mathbf{\Theta}'\|_{\max}\le\delta} (f^{1/2}(\boldsymbol{\mu}_1,\boldsymbol{y}) - f^{1/2}(\boldsymbol{\mu}_2,\boldsymbol{y}))^2 d\boldsymbol{y} \\ &\le 2 - 2\frac{1}{(\sqrt{2\pi}\sigma)^k}\int \inf_{\|\mathbf{\Theta}-\mathbf{\Theta}'\|_{\max}\le\delta} \exp\left(-\frac{\|\boldsymbol{y}-\frac{\boldsymbol{\mu}_1+\boldsymbol{\mu}_2}{2}\|_2^2 + \|\boldsymbol{\mu}_1-\boldsymbol{\mu}_2\|_2^2/2}{2\sigma^2}\right) d\boldsymbol{y} \\ &\le 2 - 2\inf_{\|\mathbf{\Theta}-\mathbf{\Theta}'\|_{\max}\le\delta}\exp\left(-\frac{\|\boldsymbol{\mu}_1-\boldsymbol{\mu}_2\|_2^2}{4\sigma^2}\right) \\ &\le \frac{(\|\boldsymbol{a}\|_1)^2\|\mathbf{\Theta}-\mathbf{\Theta}'\|_{\max}^2}{4\sigma^2} \le \frac{(\|\boldsymbol{a}\|_1)^2\delta^2}{4\sigma^2}. \end{aligned}$$

The second inequality follows from the invariance property of the normal distribution. Corollary 2 follows when $\|\boldsymbol{a}\|_1$ is bounded. This completes the proof. ■

Proof of Theorem 2: After some calculations, we obtain that

$$\begin{aligned} h^2(\boldsymbol{\Theta},\boldsymbol{\Theta}^0) &= 2\left(1-\prod_{i=1}^{n}\frac{1}{(\sqrt{2\pi}\sigma)^k}\int \exp\big[-\frac{1}{4\sigma^2}(\|\boldsymbol{y}_i-\boldsymbol{a}_i^T\boldsymbol{\Theta}\|^2+\|\boldsymbol{y}_i-\boldsymbol{a}_i^T\boldsymbol{\Theta}^0\|^2)\big]d\boldsymbol{y}\right)\\ &= 2\left(1-\prod_{i=1}^{n}\exp\big[-\frac{1}{8\sigma^2}\|\boldsymbol{a}_i^T(\boldsymbol{\Theta}-\boldsymbol{\Theta}^0)\|^2\right)\\ &= 2\left(1-\exp(-\frac{1}{8\sigma^2}\|\boldsymbol{A}(\boldsymbol{\Theta}-\boldsymbol{\Theta}^0)\|_F^2)\right),\\ K(\boldsymbol{\Theta}^0,\boldsymbol{\Theta}) &= \frac{1}{2\sigma^2}\|\boldsymbol{A}(\boldsymbol{\Theta}-\boldsymbol{\Theta}^0)\|_F^2. \end{aligned}$$

When $\epsilon<1$,

$$\begin{aligned} P(K(\boldsymbol{\Theta}^0,\hat{\boldsymbol{\Theta}}^{L_0})\geq 4\epsilon^2) &= P\left(\frac{1}{8\sigma^2}\|\boldsymbol{A}(\hat{\boldsymbol{\Theta}}^{L_0}-\boldsymbol{\Theta}^0)\|_F^2\geq\epsilon^2\right)\\ &\leq P\left(\frac{1}{8\sigma^2}\|\boldsymbol{A}(\hat{\boldsymbol{\Theta}}^{L_0}-\boldsymbol{\Theta}^0)\|_F^2\geq -\log(1-\frac{\epsilon^2}{2})\right)\\ &= P\left(2\left(1-\exp(-\frac{1}{8\sigma^2}\|\boldsymbol{A}(\hat{\boldsymbol{\Theta}}^{L_0}-\boldsymbol{\Theta}^0)\|_F^2)\right)\geq\epsilon^2\right)\\ &= P\left(h^2(\hat{\boldsymbol{\Theta}}^{L_0},\boldsymbol{\Theta}^0)\geq\epsilon^2\right). \end{aligned}$$

For any $\epsilon\geq\epsilon_{n,p,k}$, it follows from Theorem 1 and Corollary 2 that

$$\begin{aligned} EK(\boldsymbol{\Theta}^0,\hat{\boldsymbol{\Theta}}^{L_0}) &\leq EK(\boldsymbol{\Theta}^0,\hat{\boldsymbol{\Theta}}^{L_0})I\{K(\boldsymbol{\Theta}^0,\hat{\boldsymbol{\Theta}}^{L_0})\leq 4\epsilon^2\}+EK(\boldsymbol{\Theta}^0,\hat{\boldsymbol{\Theta}}^{L_0})I\{K(\boldsymbol{\Theta}^0,\hat{\boldsymbol{\Theta}}^{L_0})>4\epsilon^2\}\\ &\leq 4\epsilon^2+\Big(EK^2(\boldsymbol{\Theta}^0,\hat{\boldsymbol{\Theta}}^{L_0})\Big)^{1/2}\Big(P(K^2(\boldsymbol{\Theta}^0,\hat{\boldsymbol{\Theta}}^{L_0})>4\epsilon^2)\Big)^{1/2}. \end{aligned}$$

By the triangle inequality, $\|\boldsymbol{A}\boldsymbol{\Theta}^0-\boldsymbol{A}\hat{\boldsymbol{\Theta}}^{L_0}\|_F-\|\boldsymbol{\epsilon}\|_F\leq\|\boldsymbol{A}\boldsymbol{\Theta}^0+\boldsymbol{\epsilon}-\boldsymbol{A}\hat{\boldsymbol{\Theta}}^{L_0}\|_F$. Note that $\hat{\boldsymbol{\Theta}}^{L_0}$ is a global minimizer of (3). Then $\|\boldsymbol{A}\boldsymbol{\Theta}^0+\boldsymbol{\epsilon}-\boldsymbol{A}\hat{\boldsymbol{\Theta}}^{L_0}\|_F\leq\|\boldsymbol{\epsilon}\|_F$. Hence

$$K(\boldsymbol{\Theta}^0,\hat{\boldsymbol{\Theta}}^{L_0})=\frac{1}{2\sigma^2}\|\boldsymbol{A}(\boldsymbol{\Theta}^0-\hat{\boldsymbol{\Theta}}^{L_0})\|_F^2\leq\frac{2}{\sigma^2}\|\boldsymbol{\epsilon}\|_F^2.$$

Thus,

$$\begin{aligned} EK(\boldsymbol{\Theta}^0,\hat{\boldsymbol{\Theta}}^{L_0}) &\leq 4\epsilon^2+\Big(E\frac{4}{\sigma^4}\|\boldsymbol{\epsilon}\|_F^4\Big)^{1/2}P\big(K^2(\boldsymbol{\Theta}^0,\hat{\boldsymbol{\Theta}}^{L_0})>4\epsilon^2\big)\\ &\leq 4\epsilon^2+10\exp(-c_1n\epsilon^2+\log\sqrt{3nk}). \end{aligned}$$

The results in Theorem 2 follow by letting $\epsilon=\epsilon_{n,p,k}$ and using the fact that $\log k\leq C_{p,k}^2$. This completes the proof. ■

Proof of Theorem 3: Without loss of generality, assume $p\geq k$ and $n=p$. When $\sigma=O(1/\sqrt{p})$, by Theorem 2, we have

$$\begin{aligned} \|\hat{\boldsymbol{\Theta}}^{L_0}-\boldsymbol{\Theta}^0\|_F^2=2\sigma^2K(\boldsymbol{\Theta}^0,\hat{\boldsymbol{\Theta}}^{L_0}) &= O_P(\frac{\epsilon_{n,p,k}^2}{p})\\ &= O_P\left(\frac{C_{p,k}^2}{p^2}\log(\frac{\sqrt{p}}{C_{p,k}})\right)\\ &= O_P\left(\frac{C_{p,k}^2}{p^2}\log(\frac{p^2}{C_{p,k}^2})\right), \end{aligned}\tag{26}$$

where

$$C_{p,k} = O\left(\sqrt{\log(p)}\sqrt{(p+k)s_2^0 + s_1^0} + \sqrt{s_1^0 \log\left(e\frac{(p+k-r(\mathbf{\Theta}^0))r(\mathbf{\Theta}^0)}{s_1^0}\right)}\right). \tag{27}$$

(27) comes from the proof of Corollary 2 with M in **Assumption A** being $O(\sqrt{p})$. Thus,

$$\|\hat{\mathbf{\Theta}}^{L_0} - \mathbf{\Theta}^0\|_F^2 = O_P\left(C'_{p,k}\log(\frac{1}{C'_{p,k}})\right)$$

with

$$C'_{p,k} = \frac{\log(p)\cdot[(p+k)s_2^0 + s_1^0] + s_1^0\log\left(e\frac{(p+k-r(\mathbf{\Theta}^0))r(\mathbf{\Theta}^0)}{s_1^0}\right)}{p^2}.$$

This completes the proof. ■

References

Agarwal, A., Negahban, S. and Wainwright, M. (2012). Noisy matrix decomposition via convex relaxation: optimal rates in high dimensions. *The Annals of Statistics*, Vol. 40, No. 2, 1171–1197.

Amit, Y., Fink, M., Srebro, N., and Ullman, S. (2007). Uncovering shared structures in multiclass classification. *Proceedings of the 24th Annual International Conference on Machine learning*, 17–24.

Arlot, S. and Celisse, A. (2010). A survey of cross-validation procedures for model selection. *Statist. Surv.*, 4, 40–79.

Beck, Amir and Teboulle, Marc (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, Vol. 2, No. 1, 183-202.

Bunea, F., and She, Y. (2011). Optimal selection of reduced rank estimators of high-dimensional matrices. *Annals of Statistics*, 39(2), 1282-1309.

Cai, J.F., Candès, E.J. and Shen, Z. (2010).A singular value thresholding algorithm for matrix completion. Arxiv preprint arXiv:0810.3286.

Cai, T. T., Liu, W. and Luo, X. (2011). A constrained l_1 minimization approach to sparse precision matrix estimation. *J. Amer. Statist. Assoc.*, 106, 594-607.

Candes, E., Li. X., Ma, U., and Wright, J. (2009). Robust principal component analysis. *Journal of ACM*, 58(1), 1-37.

Candes, E.J. and Recht, B. (2009). Exact matrix completion via convex optimization. *Found of Comput. Math.*, 9, 717-772.

Chandrasekaran, V., Sanghavi, S., Parrilo, P.A., and Willsky, A.S. (2011). Rank-sparsity incoherence for matrix decomposition, *SIAM J. Optim.*, 21, 572-596.

Chen, B. , He, S., Li, Z. and Zhang, S. (2012). Maximum block improvement and polynomial optimization., *SIAM Journal on Optimization*, 22, 87-107.

Chen, J., Liu, J., and Ye, J. (2010). Learning incoherent sparse and low-rank patterns from multiple tasks. *The Sixteenth ACM SIGKDD International Conference On Knowledge Discovery and Data Mining.* (SIGKDD 2010).

Efron, B. (2004). The estimation of prediction error: Covariance penalties and cross-validation. *Journal of the American Statistical Association*, 99, 619-642. (with discussion).

Ganesh, A., Min, K., Wright, J. and Ma, Y. (2012). Principal component pursuit with reduced linear measurements. *International Symposium on Information Theory.*

Golub, G. and Van Loan, C. (1996). *Matrix Computations.* Third edition. London: The Johns Hopkins University Press.

Halko, N., Martinsson P. G. and Tropp, J. A.. (2011). Finding structure with randomness: stochastic algorithms for constructing approximate matrix decompositions. *SIAM Rev.*, 53(2), 217-288.

Jain, P., Meka, R. and Dhillon, I. (2010). Guaranteed rank minimization via singular value projection. *Advances in Neural Information Processing Systems*, 23, 937–945.

Kolmogorov, A.N. and Tihomirov, V.M. (1959). ϵ-entropy and ϵ-capacity of sets in function spaces. *Uspekhi Mat. Nauk.* **14** *3-86.* [In Russian. English translation, *Ameri. Math. Soc. Transl.* **2, 17,** 277-364.(1961)].

Lin, Z., Chen, M., Wu, L. and Ma, Y. (2009). The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices. *UIUC Technical Report UILU-ENG-09-2215.*

Liu, J., and Ye, J. (2009). Efficient Euclidean projections in linear time. *The Twenty-Sixth International Conference on Machine Learning.*

Negahban, S. and Wainwright, M.J. (2011). Estimation of (near) low-rank matrices with noise and high-dimensional scaling. *Annals of Statistics*, 39(2), 1069-1097.

Ossiander, M. (1987). A central limit theory under metric entropy with L_2 bracketing. *Ann. Probab.* **15** 897-919.

Porat, B. (1997). *A Course in Digital Signal Processing*, New York: John Wiley.

She, Y. (2013). Reduced rank vector generalized linear models for feature extraction. *Statistics and Its Interface*, Vol 6, 197-209.

Shen, X., Pan, W., and Zhu, Y. (2012). Likelihood-based selection and sharp parameter estimation. *Journal of the American Statistical Association*, 107, 223-232.

Srebro, N., Rennie, J.D.M., and Jaakkola, T.S. (2005). Maximum-margin matrix factorization. *Advances in Neural Information Processing Systems*, 17, 1329–1336.

Stanica, P., and Montgomery, A.P. (2001). Good lower and upper bounds on binomial coefficients. *J. Ineq. in Pure. Appl. Math.*, 2, art 30.

Stock, J. H. and Watson, M. W. (2012). Generalized shrinkage methods for forecasting using many predictors. *Journal of Business and Economic Statistics.*

Zhou, T. and Tao, D. (2011). GoDec: Randomized low-rank & sparse matrix decomposition in noisy case. *Proceedings of the 28th International Conference on Machine Learning*, Bellevue, WA, USA.

Waters, A.E., Sankaranarayanan, A.C. and Baraniuk, R.G.(2011). SpaRCS: Recovering low-rank and sparse matrices from compressive measurements. *Neural Information Processing Systems*, Granada, Spain.

Wong, W.H., and Shen, X. (1995). Probability inequalities for likelihood ratios and convergence rates of sieve MLEs. *Ann. Statist.*, 23, 339-362.

Wright, J., Ganesh, A., Min, K., and Ma, Y. (2013). Compressive principal component pursuit. *Information and Inference*.

Wright, J., Ganesh, A., Rao, S., and Ma, Y. (2009). Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. *Advances in Neural Information Processing Systems*.

Xing, S., Zhu, Y., Shen, X., and Ye, J. (2012). Optimal exact rank minimization for noisy data. *Proceeding the 18th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Beijing, China.

Yuan, X. and Yang, J. (2013). Sparse and low-rank matrix decomposition via alternating direction methods. *Pacific Journal of Optimization*, 9(1), 167-180.

Journal of Machine Learning Research 16 (2015) 77-102 Submitted 7/14; Published 1/15

Statistical Topological Data Analysis using Persistence Landscapes

Peter Bubenik PETER.BUBENIK@GMAIL.COM
Department of Mathematics
Cleveland State University
Cleveland, OH 44115-2214, USA

Editor: David Dunson

Abstract

We define a new topological summary for data that we call the persistence landscape. Since this summary lies in a vector space, it is easy to combine with tools from statistics and machine learning, in contrast to the standard topological summaries. Viewed as a random variable with values in a Banach space, this summary obeys a strong law of large numbers and a central limit theorem. We show how a number of standard statistical tests can be used for statistical inference using this summary. We also prove that this summary is stable and that it can be used to provide lower bounds for the bottleneck and Wasserstein distances.

Keywords: topological data analysis, statistical topology, persistent homology, topological summary, persistence landscape

1. Introduction

Topological data analysis (TDA) consists of a growing set of methods that provide insight to the "shape" of data (see the surveys Ghrist, 2008; Carlsson, 2009). These tools may be of particular use in understanding global features of high dimensional data that are not readily accessible using other techniques. The use of TDA has been limited by the difficulty of combining the main tool of the subject, the *barcode* or *persistence diagram* with statistics and machine learning. Here we present an alternative approach, using a new summary that we call the *persistence landscape*. The main technical advantage of this descriptor is that it is a function and so we can use the vector space structure of its underlying function space. In fact, this function space is a separable Banach space and we apply the theory of random variables with values in such spaces. Furthermore, since the persistence landscapes are sequences of piecewise-linear functions, calculations with them are much faster than the corresponding calculations with barcodes or persistence diagrams, removing a second serious obstruction to the wider use of topological methods in data analysis.

Notable successes of TDA include the discovery of a subgroup of breast cancers by Nicolau et al. (2011), an understanding of the topology of the space of natural images by Carlsson et al. (2008) and the topology of orthodontic data by Heo et al. (2012), and the detection of genes with a periodic profile by Dequéant et al. (2008). De Silva and Ghrist (2007b,a) used topology to prove coverage in sensor networks.

In the standard paradigm for TDA, one starts with data that one encodes as a finite set of points in $\mathbb{R}^n$ or more generally in some metric space. Then one applies some geometric construction to which one applies tools from algebraic topology. The end result is a topological summary of the data. The standard topological descriptors are the barcode and the persistence diagram (Edelsbrunner et al., 2002; Zomorodian and Carlsson, 2005; Cohen-Steiner et al., 2007), which give a multiscale representation of the *homology* (Hatcher, 2002) of the geometric construction. Roughly, homology in degree 0 describes the connectedness of the data; homology in degree 1 detects holes or tunnels; homology in degree 2 captures voids; and so on. Of particular interest are the homological features that persist as the resolution changes. We will give precise definitions and an illustrative example of this method, called *persistent homology* or *topological persistence*, in Section 2.

Now let us take a statistical view of this paradigm. We consider the data to be sampled from some underlying abstract probability space. Composing the constructions above, we consider our topological summary to be a random variable with values in some summary space $\mathcal{S}$. In detail, the probability space $(\Omega, \mathcal{F}, \mathcal{P})$ consists of a sample space Ω, a σ-algebra $\mathcal{F}$ of events, and a probability measure $\mathcal{P}$. Composing our constructions gives a function $X : (\Omega, \mathcal{F}, \mathcal{P}) \to (\mathcal{S}, \mathcal{A}, \mathcal{P}_*)$, where $\mathcal{S}$ is the summary space, which we assume has some metric, $\mathcal{A}$ is the corresponding Borel σ-algebra, and $\mathcal{P}_*$ is the probability measure on $\mathcal{S}$ obtained by pushing forward $\mathcal{P}$ along X. We assume that X is measurable and thus X is a random variable with values in $\mathcal{S}$.

Here is a list of what we would like to be able to do with our topological summary. Let $X_1, \ldots, X_n$ be a sample of independent random variables with the same distribution as X. We would like to have a good notion of the mean μ of X and the mean $\overline{X}_n$ of the sample; know that $\overline{X}_n$ converges to μ; and be able to calculate $\overline{X}_n(\omega)$, for $\omega \in \Omega$, efficiently. We would like to have information the difference $\overline{X}_n - \mu$, and be able to calculate approximate confidence intervals related to μ. Given two such samples for random variables X and Y with values in our summary space, we would like to be able to test the hypothesis that $\mu_X = \mu_Y$. In order to answer these questions we also need an efficient algorithm for calculating distances between elements of our summary space. In this article, we construct a topological summary that we call the persistence landscape which meets these requirements.

Our basic idea is to convert the barcode into a function in a somewhat additive manner. The are many possible variations of this construction that may result in more suitable summary statistics for certain applications. Hopefully, the theory presented here will also be helpful in those situations.

We remark that while the persistence landscape has a corresponding barcode and persistence diagram, the mean persistence landscape does not. This is analogous to the situation in which an integer-valued random variable having a Poisson distribution has a summary statistic, the rate parameter, that is not an integer.

We also remark that the reader may restrict our Banach space results results to the perhaps more familiar Hilbert space setting. However we will need this generality to prove stability of the persistence landscape for, say, functions on the n-dimensional sphere where $n > 2$.

There has been progress towards combining the persistence diagram and statistics (Mileyko et al., 2011; Turner et al., 2014; Munch et al., 2013; Chazal et al., 2013; Fasy et al., 2014). Blumberg et al. (2014) give a related statistical approach to TDA. Kovacev-Nikolic

et al. (2014) use the persistence landscape defined here to study the maltose binding complex and Chazal et al. (2014) apply the bootstrap to the persistence landscape. The persistence landscape is related to the well group defined by Edelsbrunner et al. (2011).

In Section 2 we provide the necessary background and define the persistence landscape and give some of its properties. In Section 3 we introduce the statistical theory of persistence landscapes, which we apply to a few examples in Section 4. In Section 5 we prove that the persistence landscape is stable and that it provides lower bounds for the previously defined bottleneck and Wasserstein distances.

2. Topological Summaries

The two standard topological summaries of data are the *barcode* and the *persistence diagram*. We will define a new closely-related summary, the *persistence landscape*, and then compare it to these two previous summaries. All of these summaries are derived from the *persistence module*, which we now define.

2.1 Persistence Modules

The main algebraic object of study in topological data analysis is the persistence module. A *persistence module* M consists of a vector space M_a for all $a \in \mathbb{R}$ and linear maps $M(a \leq b) : M_a \to M_b$ for all $a \leq b$ such that $M(a \leq a)$ is the identity map and for all $a \leq b \leq c$, $M(b \leq c) \circ M(a \leq b) = M(a \leq c)$.

There are many ways of constructing a persistence module. One example starts with a set of points $X = \{x_1, \ldots, x_n\}$ in the plane $M = \mathbb{R}^2$ as shown in the top left of Figure 1. To help understand this configuration, we "thicken" each point, by replacing each point, x, with $B_x(r) = \{y \in M \mid d(x, y) \leq r\}$, a disk of fixed radius, r, centered at x. The resulting union, $X_r = \bigcup_{i=1}^n B_r(x_i)$, is shown in Figure 1 for various values of r. For each r, we can calculate $H(X_r)$, the homology of the resulting union of disks. To be precise, $H(-)$ denotes $H_k(-, \mathbb{F})$, the singular homology functor in degree k with coefficients in a field $\mathbb{F}$. So $H(X_r)$ is a vector space that is the quotient of the k-cycles modulo those that are boundaries. As r increases, the union of disks grows, and the resulting inclusions induce maps between the corresponding homology groups. More precisely, if $r \leq s$, the inclusion $\iota_r^s : X_r \hookrightarrow X_s$ induces a map $H(\iota_r^s) : H(X_r) \to H(X_s)$. The images of these maps are the *persistent homology* groups. The collection of vector spaces $H(X_r)$ and linear maps $H(\iota_r^s)$ is a persistence module. Note that this construction works for any set of points in $\mathbb{R}^n$ or more generally in a metric space.

The union of balls X_r has a nice combinatorial description. The *Čech complex*, $\check{C}_r(X)$, of the set of balls $\{B_{x_i}(r)\}$ is the simplicial complex whose vertices are the points $\{x_i\}$ and whose k-simplices correspond to $k+1$ balls with nonempty intersection (see Figure 1). This is also called the *nerve*. It is a basic result that if the ambient space is $\mathbb{R}^n$, X_r is homotopy equivalent to its Čech complex (Borsuk, 1948). So to obtain the singular homology of the union of balls, one can calculate the simplicial homology of the corresponding Čech complex. The Čech complexes $\{\check{C}_r(X)\}$ together with the inclusions $\check{C}_r(X) \subseteq \check{C}_s(X)$ for $r \leq s$ form a filtered simplicial complex. Applying simplicial homology we obtain a persistence module. There exist efficient algorithms for calculating the persistent homology of filtered simplicial complexes (Edelsbrunner et al., 2002; Milosavljević et al., 2011; Chen and Kerber, 2013).

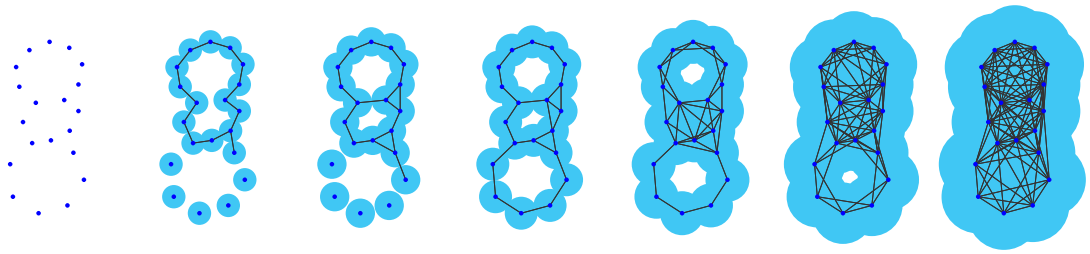


Figure 1: A growing union of balls and the 1-skeleton of the corresponding Čech complex. As the radius grows, features—such as connected components and holes—appear and disappear. Here, the complexes illustrate the births and deaths of three holes, homology classes in degree one. The corresponding birth-death pairs are plotted as part of the top left of Figure 2.

The Čech complex is often computationally expensive, so many variants have been used in computational topology. A larger, but simpler complex called the Rips complex has as vertices the points x_i and has k-simplices corresponding to $k+1$ balls with all pairwise intersections nonempty. Other possibilities include the witness complexes of de Silva and Carlsson (2004), graph induced complexes by Dey et al. (2013) and complexes built using kernel density estimators and triangulations of the ambient space (Bubenik et al., 2010). Some of these are used in the examples in Section 4.

Given any real-valued function $f : S \to \mathbb{R}$ on a topological space S, we can define the associated persistence module, $M(f)$, where $M(f)(a) = H(f^{-1}((\infty, a]))$ and $M(f)(a \leq b)$ is induced by inclusion. Taking f to be the the minimum distance to a finite set of points, X, we obtain the first example.

2.2 Persistence Landscapes

In this section we define a number of functions derived from a persistence module. Examples of each of these are given in Figure 2.

Let M be a persistence module. For $a \leq b$, the corresponding *Betti number* of M, is given by the dimension of the image of the corresponding linear map. That is,

$$\beta^{a,b} = \dim(\mathrm{im}(M(a \leq b))). \tag{1}$$

Lemma 1 *If $a \leq b \leq c \leq d$ then $\beta^{b,c} \geq \beta^{a,d}$.*

Proof Since $M(a \leq d) = M(c \leq d) \circ M(b \leq c) \circ M(a \leq b)$, this follows from (1). ■

Our simplest function, which we call the *rank function* is the function $\lambda : \mathbb{R}^2 \to \mathbb{R}$ given by

$$\lambda(b, d) = \begin{cases} \beta^{b,d} & \text{if } b \leq d \\ 0 & \text{otherwise.} \end{cases}$$

Now let us change coordinates so that the resulting function is supported on the upper half plane. Let

$$m = \frac{b+d}{2}, \quad \text{and} \quad h = \frac{d-b}{2}. \tag{2}$$

The *rescaled rank function* is the function $\lambda : \mathbb{R}^2 \to \mathbb{R}$ given by

$$\lambda(m,h) = \begin{cases} \beta^{m-h,m+h} & \text{if } h \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$

Much of our theory will apply to these simple functions. However, the following version, which we will call the *persistence landscape*, will have some advantages.

First let us observe that for a fixed $t \in \mathbb{R}$, $\beta^{t-\bullet,t+\bullet}$ is a decreasing function. That is,

Lemma 2 *For* $0 \leq h_1 \leq h_2$,

$$\beta^{t-h_1,t+h_1} \geq \beta^{t-h_2,t+h_2}.$$

Proof Since $t - h_2 \leq t - h_1 \leq t + h_1 \leq t + h_2$, by Lemma 1, $\beta^{t-h_2,t+h_2} \leq \beta^{t-h_1,t+h_1}$. ■

Definition 3 *The* persistence landscape *is a function* $\lambda : \mathbb{N} \times \mathbb{R} \to \overline{\mathbb{R}}$, *where* $\overline{\mathbb{R}}$ *denotes the extended real numbers,* $[-\infty, \infty]$. *Alternatively, it may be thought of as a sequence of functions* $\lambda_k : \mathbb{R} \to \overline{\mathbb{R}}$, *where* $\lambda_k(t) = \lambda(k,t)$. *Define*

$$\lambda_k(t) = \sup(m \geq 0 \mid \beta^{t-m,t+m} \geq k).$$

The persistence landscape has the following properties.

Lemma 4 *1.* $\lambda_k(t) \geq 0$,

2. $\lambda_k(t) \geq \lambda_{k+1}(t)$, *and*

3. λ_k *is 1-Lipschitz.*

The first two properties follow directly from the definition. We prove the third in the appendix.

To help visualize the graph of $\lambda : \mathbb{N} \times \mathbb{R} \to \overline{\mathbb{R}}$, we can extend it to a function $\overline{\lambda} : \mathbb{R}^2 \to \overline{\mathbb{R}}$ by setting

$$\overline{\lambda}(x,t) = \begin{cases} \lambda(\lceil x \rceil, t), & \text{if } x > 0, \\ 0, & \text{if } x \leq 0. \end{cases} \tag{3}$$

We remark that the non-persistent Betti numbers, $\{\dim(M(t))\}$, of a persistence module M can be read off from the diagonal of the rank function, the m-axis of the rescaled rank function, and from the support of the persistence landscape.

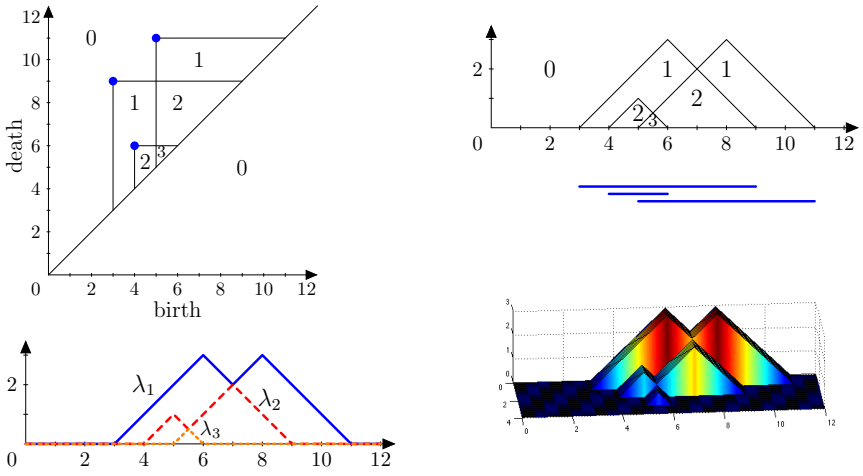


Figure 2: Persistence landscapes for the homology in degree 1 of the example in Figure 1. For the rank function (top left) and rescaled rank function (top right) the values of the functions on the corresponding region are given. The top left graph also contains the three points of the corresponding persistence diagram. Below the top right graph is the corresponding barcode. We also have the corresponding persistence landscape (bottom left) and its 3d-version (bottom right). Notice that λ_1 gives a measure of the dominant homological feature at each point of the filtration.

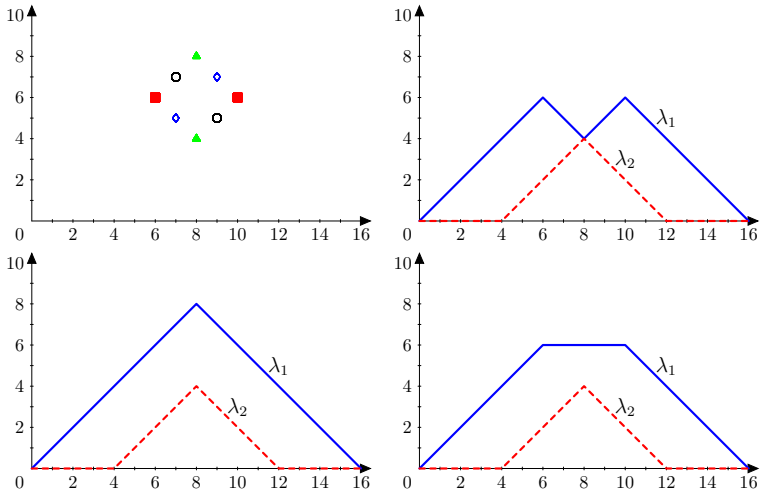


Figure 3: Means of persistence diagrams and persistence landscapes. Top left: the rescaled persistence diagrams $\{(6,6),(10,6)\}$ and $\{(8,4),(8,8)\}$ have two (Fréchet) means: $\{(7,5),(9,7)\}$ and $\{(7,7),(9,5)\}$. In contrast their corresponding persistence landscapes (top right and bottom left) have a unique mean (bottom right).

2.3 Barcodes and Persistence Diagrams

All of the information in a (tame) persistence module is completely contained in a multiset of intervals called a *barcode* (Zomorodian and Carlsson, 2005; Crawley-Boevey, 2012; Chazal et al., 2012). Mapping each interval to its endpoints we obtain the *persistence diagram*.

There exist maps in both directions between these topological summaries and our functions. For an example of corresponding persistence diagrams, barcodes and persistence landscapes, see Figure 2. Informally, the persistence diagram consists of the "upper-left corners" in our rank function. In the other direction, $\lambda(b,d)$ counts the number of points in the persistence diagram in the upper left quadrant of (b,d). Informally, the barcode consists of the "bases of the triangles" in the rescaled rank function, and the other direction is obtained by "stacking isosceles triangles" whose bases are the intervals in the barcode. We invite the reader to make the mappings precise. For example, given a persistence diagram $\{(b_i,d_i)\}_{i=1}^n$,

$$\lambda_k(t) = k\text{th largest value of } \min(t-b_i, d_i-t)_+,$$

where c_+ denotes $\max(c,0)$. The fact that barcodes are a complete invariant of persistence modules is central to these equivalences.

The geometry of the space of persistence diagrams makes it hard to work with. For example, sets of persistence diagrams need not have a unique (Fréchet) mean (Mileyko et al., 2011). In contrast, the space of persistence landscapes is very nice. So a set of persistence landscapes has a unique mean (4). See Figure 3.

Compared to the persistence diagram, the barcode has extra information on whether or not the endpoints of the intervals are included. This finer information is seen in the rank

function and rescaled rank function, but not in the persistence landscape. However when we pass to the corresponding L^p space in Section 2.4, this information disappears.

2.4 Norms for Persistence Landscapes

Recall that for a measure space $(\mathcal{S}, \mathcal{A}, \mu)$, and a function $f : \mathcal{S} \to \mathbb{R}$ defined μ-almost everywhere, for $1 \leq p < \infty$, $\|f\|_p = \left[\int |f|^p d\mu\right]^{\frac{1}{p}}$, and $\|f\|_\infty = \operatorname{ess\,sup} f = \inf\{a \mid \mu\{s \in \mathcal{S} \mid f(s) > a\} = 0\}$. For $1 \leq p \leq \infty$, $\mathcal{L}^p(\mathcal{S}) = \{f : \mathcal{S} \to \mathbb{R} \mid \|f\|_p < \infty\}$ and define $L^p(\mathcal{S}) = \mathcal{L}^p(\mathcal{S})/\sim$, where $f \sim g$ if $\|f - g\|_p = 0$.

On $\mathbb{R}$ and $\mathbb{R}^2$ we will use the Lebesgue measure. On $\mathbb{N} \times \mathbb{R}$, we use the product of the counting measure on $\mathbb{N}$ and the Lebesgue measure on $\mathbb{R}$. For $1 \leq p < \infty$ and $\lambda : \mathbb{N} \times \mathbb{R} \to \overline{\mathbb{R}}$,

$$\|\lambda\|_p^p = \sum_{k=1}^{\infty} \|\lambda_k\|_p^p,$$

where $\lambda_k(t) = \lambda(k, t)$. By Lemma 4(2), $\|\lambda\|_\infty = \|\lambda_1\|_\infty$. If we extend f to $\overline{\lambda} : \mathbb{R}^2 \to \overline{\mathbb{R}}$, as in (3), we have $\|\lambda\|_p = \|\overline{\lambda}\|_p$, for $1 \leq p \leq \infty$.

If λ is any of our functions corresponding to a barcode that is a finite collection of finite intervals, then $\lambda \in \mathcal{L}^p(\mathcal{S})$ for $1 \leq p \leq \infty$, where $\mathcal{S}$ equals $\mathbb{N} \times \mathbb{R}$ or $\mathbb{R}^2$.

Let λ_{bd} and λ_{mh} denote the rank function and the rescaled rank function corresponding to a persistence landscape λ, and let D be the corresponding persistence diagram. Let $\operatorname{pers}_2(D)$ denote the sum of the squares of the lengths of the intervals in the corresponding barcode, and let $\operatorname{pers}_\infty(D)$ be the length of the longest interval.

Proposition 5 *1.* $\|\lambda\|_1 = \|\lambda_{mh}\|_1 = \frac{1}{2}\|\lambda_{bd}\|_1 = \frac{1}{4}\operatorname{pers}_2(D)$, *and*

2. $\|\lambda\|_\infty = \|\lambda_1\|_\infty = \frac{1}{2}\operatorname{pers}_\infty(D)$.

Proof

1. To see that $\|\lambda\|_1 = \|\lambda_{mh}\|$ we remark that both are the volume of the same solid. The change of coordinates implies that $\|\lambda_{mh}\|_1 = \frac{1}{2}\|\lambda_{bd}\|_1$. If $D = \{(b_i, d_i)\}$, then each point (b_i, d_i) contributes h_i^2 to the volume $\|\lambda_{mh}\|_1$, where $h_i = \frac{d_i - b_i}{2}$. So $\|\lambda_{mh}\|_1 = \sum_i h_i^2$. Finally, $\operatorname{pers}_2(D) = \sum_i (2h_i)^2 = 4\sum_i h_i^2$.
2. Lemma 4(2) implies that $\|\lambda\|_\infty = \|\lambda_1\|_\infty$. If $D = \{(b_i, d_i)\}$, then $\|\lambda\|_\infty = \sup_i \frac{d_i - b_i}{2}$.

■

We remark that the quantities in 1 and 2 also equal $W_2(D, \emptyset)^2$ and $W_\infty(D, \emptyset)$ respectively (see Section 5 for the corresponding definitions).

3. Statistics with Landscapes

Now let us take a probabilistic viewpoint. First, we assume that our persistence landscapes lie in $L^p(\mathcal{S})$ for some $1 \leq p < \infty$, where $\mathcal{S}$ equals $\mathbb{N} \times \mathbb{R}$ or $\mathbb{R}^2$. In this case, $L^p(\mathcal{S})$ is a separable Banach space. When $p = 2$ we have a Hilbert space; however, we will not use this structure. In some examples, the persistence landscapes will only be stable for some $p > 2$ (see Theorem 16).

3.1 Landscapes as Banach Space Valued Random Variables

Let X be a random variable on some underlying probability space $(\Omega, \mathcal{F}, P)$, with corresponding persistence landscape Λ, a Borel random variable with values in the separable Banach space $L^p(\mathcal{S})$. That is, for $\omega \in \Omega$, $X(\omega)$ is the data and $\Lambda(\omega) = \lambda(X(\omega)) =: \lambda$ is the corresponding topological summary statistic.

Now let $X_1, \ldots, X_n$ be independent and identically distributed copies of X, and let $\Lambda^1, \ldots, \Lambda^n$ be the corresponding persistence landscapes. Using the vector space structure of $L^p(\mathcal{S})$, the *mean landscape* $\overline{\Lambda}^n$ is given by the pointwise mean. That is, $\overline{\Lambda}^n(\omega) = \overline{\lambda}^n$, where

$$\overline{\lambda}^n(k,t) = \frac{1}{n}\sum_{i=1}^{n} \lambda^i(k,t). \tag{4}$$

Let us interpret the mean landscape. If $B_1, \ldots, B_n$ are the barcodes corresponding to the persistence landscapes $\lambda^1, \ldots, \lambda^n$, then for $k \in \mathbb{N}$ and $t \in \mathbb{R}$, $\overline{\lambda}^n(k,t)$ is the average value of the largest radius interval centered at t that is contained in k intervals in the barcodes $B_1, \ldots, B_n$.

For those used to working with persistence diagrams, it is tempting to try to find a persistence diagram whose persistence landscape is closest to a given mean landscape. While this is an interesting mathematical question, we would like to suggest that the more important practical issue is using the mean landscape to understand the data.

We would like to be able to say that the mean landscape converges to the expected persistence landscape. To say this precisely we need some notions from probability in Banach spaces.

3.2 Probability in Banach Spaces

Here we present some results from probability in Banach spaces. For a more detailed exposition we refer the reader to Ledoux and Talagrand (2011).

Let $\mathcal{B}$ be a real separable Banach space with norm $\|\cdot\|$. Let $(\Omega, \mathcal{F}, P)$ be a probability space, and let $V : (\Omega, \mathcal{F}, P) \to \mathcal{B}$ be a Borel random variable with values in $\mathcal{B}$. The composite $\|V\| : \Omega \xrightarrow{V} \mathcal{B} \xrightarrow{\|\cdot\|} \mathbb{R}$ is a real-valued random variable. Let $\mathcal{B}^*$ denote the topological dual space of continuous linear real-valued functions on $\mathcal{B}$. For $f \in \mathcal{B}^*$, the composite $f(V) : \Omega \xrightarrow{V} \mathcal{B} \xrightarrow{f} \mathbb{R}$ is a real-valued random variable.

For a real-valued random variable $Y : (\Omega, \mathcal{F}, P) \to \mathbb{R}$, the *mean* or *expected value*, is given by $E(Y) = \int Y \, dP = \int_\Omega Y(\omega) \, dP(\omega)$. We call an element $E(V) \in \mathcal{B}$ the *Pettis integral* of V if $E(f(V)) = f(E(V))$ for all $f \in \mathcal{B}^*$.

Proposition 6 *If $E\|V\| < \infty$, then V has a Pettis integral and $\|E(V)\| \leq E\|V\|$.*

Now let $(V_n)_{n\in\mathbb{N}}$ be a sequence of independent copies of V. For each $n \geq 1$, let $S_n = V_1 + \cdots + V_n$. For a sequence (Y_n) of $\mathcal{B}$-valued random variables, we say that (Y_n) *converges almost surely* to a $\mathcal{B}$-valued random variable Y, if $P(\lim_{n\to\infty} Y_n = Y) = 1$.

Theorem 7 (Strong Law of Large Numbers) *$(\frac{1}{n}S_n) \to E(V)$ almost surely if and only if $E\|V\| < \infty$.*

For a sequence (Y_n) of $\mathcal{B}$-valued random variables, we say that (Y_n) *converges weakly* to a $\mathcal{B}$-valued random variable Y, if $\lim_{n\to\infty} E(\varphi(Y_n)) = E(\varphi(Y))$ for all bounded continuous functions $\varphi : \mathcal{B} \to \mathbb{R}$. A random variable G with values in $\mathcal{B}$ is said to be *Gaussian* if for each $f \in \mathcal{B}^*$, $f(G)$ is a real valued Gaussian random variable with mean zero. The *covariance structure* of a $\mathcal{B}$-valued random variable, V, is given by the expectations $E[(f(V) - E(f(V)))(g(V) - E(g(V)))]$, where $f, g \in B^*$. A Gaussian random variable is determined by its covariance structure. From Hoffmann-Jørgensen and Pisier (1976) we have the following.

Theorem 8 (Central Limit Theorem) *Assume that $\mathcal{B}$ has type 2. (For example $\mathcal{B} = L^p(\mathcal{S})$, with $2 \leq p < \infty$.) If $E(V) = 0$ and $E(\|V\|^2) < \infty$ then $\frac{1}{\sqrt{n}} S_n$ converges weakly to a Gaussian random variable $G(V)$ with the same covariance structure as V.*

3.3 Convergence of Persistence Landscapes

Now we will apply the results of the previous section to persistence landscapes.

Theorem 7 directly implies the following.

Theorem 9 (Strong Law of Large Numbers for persistence landscapes) *$\overline{\Lambda}^n \to E(\Lambda)$ almost surely if and only if $E\|\Lambda\| < \infty$.*

Theorem 10 (Central Limit Theorem for peristence landscapes) *Assume $p \geq 2$. If $E\|\Lambda\| < \infty$ and $E(\|\Lambda\|^2) < \infty$ then $\sqrt{n}[\overline{\Lambda}^n - E(\Lambda)]$ converges weakly to a Gaussian random variable with the same covariance structure as Λ.*

Proof Apply Theorem 8 to $V = \lambda(X) - E(\lambda(X))$. ■

Next we apply a functional to the persistence landscapes to obtain a real-valued random variable that satisfies the usual central limit theorem.

Corollary 11 *Assume $p \geq 2$, $E\|\Lambda\| < \infty$ and $E(\|\Lambda\|^2) < \infty$. For any $f \in L^q(\mathcal{S})$ with $\frac{1}{p} + \frac{1}{q} = 1$, let*

$$Y = \int_{\mathcal{S}} f\Lambda = \|f\Lambda\|_1. \tag{5}$$

Then

$$\sqrt{n}[\overline{Y}_n - E(Y)] \xrightarrow{d} N(0, \mathrm{Var}(Y)). \tag{6}$$

where d denotes convergence in distribution and $N(\mu, \sigma^2)$ is the normal distribution with mean μ and variance σ^2.

Proof Since $V = \Lambda - E(\Lambda)$ satisfies the central limit theorem in $L^p(\mathcal{S})$, for any $g \in L^p(\mathcal{S})^*$, the real random variable $g(V)$ satisfies the central limit theorem in $\mathbb{R}$ with limiting Gaussian law with mean 0 and variance $E(g(V)^2)$. If we take $g(h) = \int_{\mathcal{S}} fh$, where $f \in L^q(\mathcal{S})$, with $\frac{1}{p} + \frac{1}{q} = 1$, then $g(V) = Y - E(Y)$ and $E(g(V)^2) = \mathrm{Var}(Y)$. ■

3.4 Confidence Intervals

The results of Section 3.3 allow us to obtain approximate confidence intervals for the expected values of functionals on persistence landscapes.

Assume that $\lambda(X)$ satisfies the conditions of Corollary 11 and that Y is a corresponding real random variable as defined in (5). By Corollary 11 and Slutsky's theorem we may use the normal distribution to obtain the approximate $(1-\alpha)$ confidence interval for $E(Y)$ using

$$\overline{Y}_n \pm z^* \frac{S_n}{\sqrt{n}},$$

where $S_n^2 = \frac{1}{n-1}\sum_{i=1}^n (Y_i - \overline{Y}_n)^2$, and z^* is the upper $\frac{\alpha}{2}$ critical value for the normal distribution.

3.5 Statistical Inference using Landscapes I

Here we apply the results of Section 3.3 to hypothesis testing using persistence landscapes.

Let $X_1, \ldots, X_n$ be an iid copies of the random variable X and let $X'_1, \ldots, X'_{n'}$ be an iid copies of the random variable X'. Assume that the corresponding persistence landscapes Λ, Λ' lie in $L^p(\mathcal{S})$, where $p \geq 2$. Let $f \in L^q(\mathcal{S})$, where $\frac{1}{p} + \frac{1}{q} = 1$. Let Y and Y' be defined as in (5). Let $\mu = E(Y)$ and $\mu' = E(Y')$. We will test the null hypothesis that $\mu = \mu'$. First we recall that the sample mean $\overline{Y} = \frac{1}{n}\sum_{i=1}^n Y_i$ is an unbiased estimator of μ and the sample variance $s_Y^2 = \frac{1}{n-1}\sum_{i=1}^n (Y_i - \overline{Y})^2$ is an unbiased estimator of $\mathrm{Var}(Y)$ and similarly for $\overline{Y'}$ and $s_{Y'}^2$. By Corollary 11, Y and Y' are asymptotically normal.

We use the two-sample z-test. Let

$$z = \frac{\overline{Y} - \overline{Y'}}{\sqrt{\frac{S_Y^2}{n} + \frac{S_{Y'}^2}{n'}}},$$

where the denominator is the standard error for the difference. From this standard score a p-value may be obtained from the normal distribution.

3.6 Choosing a Functional

To apply the above results, one needs to choose a functional, $f \in L^q(\mathcal{S})$. This choice will need to be made with an understanding of the data at hand. Here we present a couple of options.

If each $\lambda = \Lambda(\omega)$ is supported by $\{1, \ldots, K\} \times [-B, B]$, take

$$f(k,t) = \begin{cases} 1 & \text{if } t \in [-B,B] \text{ and } k \leq K \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

Then $\|f\Lambda\|_1 = \|\Lambda\|_1$.

If the parameter values for which the persistence landscape is nonzero are bounded by $\pm B$, then we have a nice choice of functional for the persistence landscape that is unavailable for the (rescaled) rank function. We can choose a functional that is sensitive of the first K dominant homological features. That is, using f in (7), $\|f\lambda\|_1 = \sum_{k=1}^K \|\Lambda_k\|_1$.

Under this weaker assumption we can also take $f_k(t) = \frac{1}{k^r}\chi_{[-B,B]}$, where $r > 1$. Then $\|f\Lambda\|_1 = \sum_{k=1}^{\infty} \frac{1}{k^r}\|\Lambda_k(t)\|_1$.

The condition that λ is supported by $\mathbb{N}\times[-B,B]$ can often be enforced by using reduced homology or by applying extended persistence (Cohen-Steiner et al., 2009; Bubenik and Scott, 2014) or by simply truncating the intervals in the corresponding barcode at some fixed values. We remark that certain experimental data may have bounds on the number of intervals. For example, in the protein data considered using the ideas presented here in Kovacev-Nikolic et al. (2014), the simplicial complexes have a fixed number of vertices.

3.7 Statistical Inference using Landscapes II

The functionals suggested in Section 3.6 in the hypothesis test given in Section 3.5 may not have enough power to discriminate between two groups with different persistence in some examples.

To increase the power, one can apply a vector of functionals and then apply Hotelling's T^2 test. For example, consider $Y = (\int(\Lambda_1 - \Lambda_1'), \ldots, \int(\Lambda_K - \Lambda_K'))$, where $K \ll n_1 + n_2 - 2$.

This alternative will not be sufficient if the persistence landscapes are translates of each other, (see Figure 7). An additional approach is to compute the distance between the mean landscapes of the two groups and obtain a p-value using a permutation test. This is done in the Section 4.3. This test has been applied to persistence diagrams and barcodes (Chung et al., 2009; Robinson and Turner, 2013).

4. Examples

The persistent homologies in this section were calculated using javaPlex (Tausz et al., 2011) and Perseus by Nanda (2013). Another publicly available alternative is Dionysus by Morozov (2012). In Section 4.2 we use Matlab code courtesy of Eliran Subag that implements an algorithm from Wood and Chan (1994).

4.1 Linked Annuli

We start with a simple example to illustrate the techniques. Following Munch et al. (2013), we sample 200 points from the uniform distribution on the union of two annuli. We then calculate the corresponding persistence landscape in degree one using the Vietoris-Rips complex. We repeat this 100 times and calculate the mean persistence landscape. See Figure 4.

Note that in the degree one barcode of this example, it is very likely that there will be one large interval, one smaller interval born at around the same time, and all other intervals are smaller and die around the time the larger two intervals are born.

4.2 Gaussian Random Fields

The topology of Gaussian random fields is of interest in statistics. The Euler characteristic of superlevel sets of a Gaussian random field may be calculated using the Gaussian Kinematic Formula of Adler and Taylor (2007). The persistent homology of Gaussian random fields

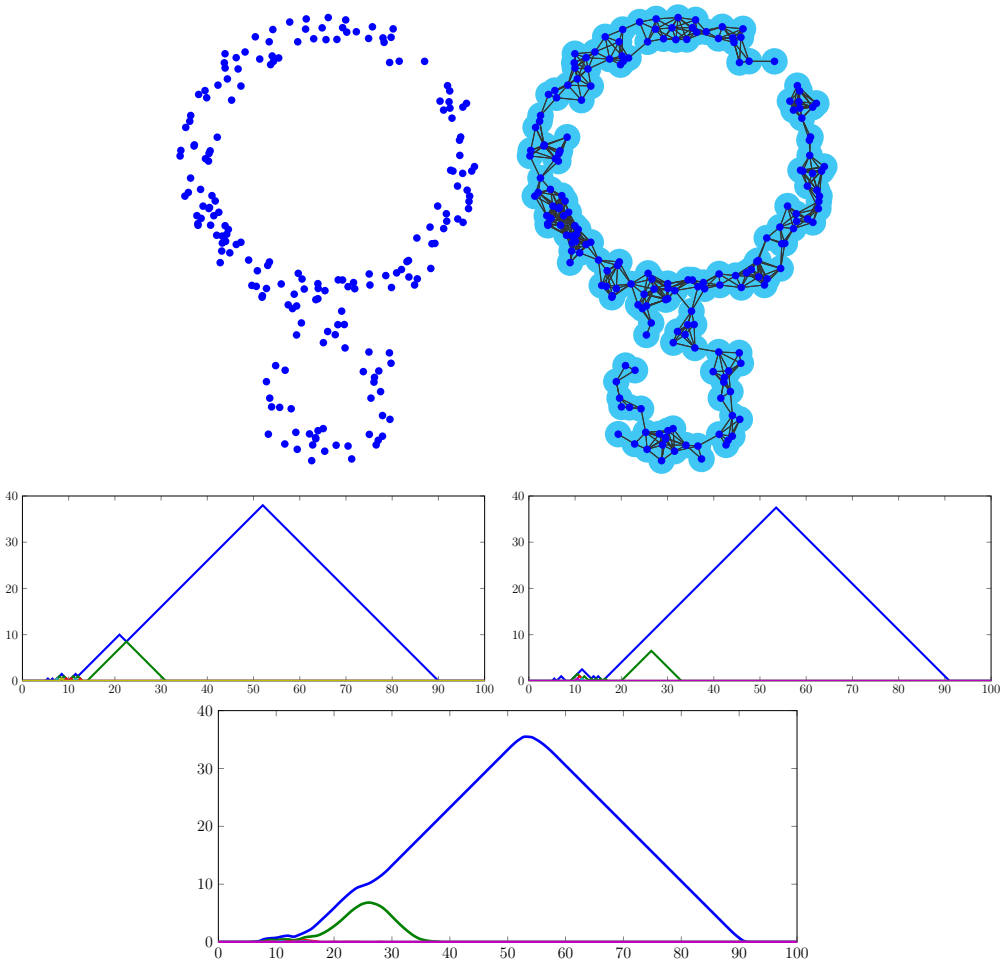


Figure 4: 200 points were sampled from a pair of linked annuli. Here we show the points and a corresponding union of balls and 1-skeleton of the Čech complex. This was repeated 100 times. Next we show two of the degree one persistence landscapes and the mean degree one persistence landscape.

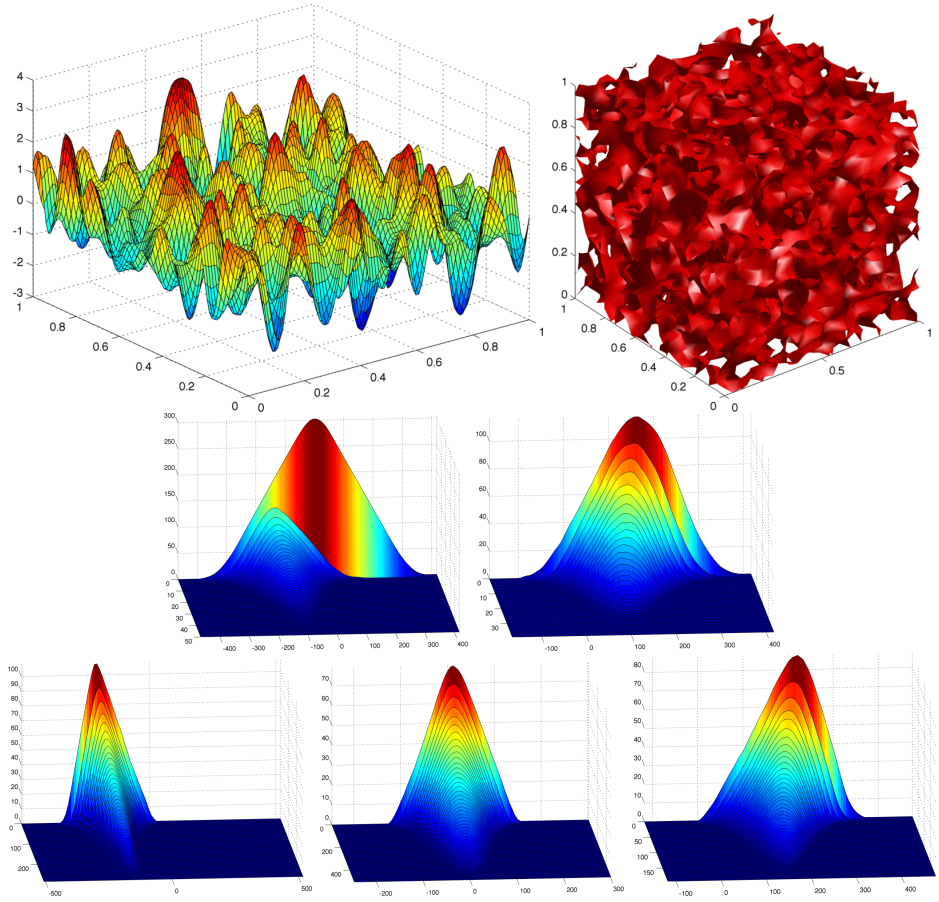


Figure 5: Mean landscapes of Gaussian random fields. The graph of a Gaussian random field on $[0,1]^2$ (top left) and its corresponding mean landscapes (middle row) in degrees 0 and 1. The 0-isosurface of a Gaussian random field on $[0,1]^3$ (top right) and the corresponding mean landscapes in degrees 0, 1 and 2 (bottom row).

has been considered by Adler et al. (2010) and its expected Euler characteristic has been obtained by Bobrowski and Borman (2012).

Here we consider a stationary Gaussian random field on $[0,1]^2$ with autocovariance function $\gamma(x,y) = e^{-400(x^2+y^2)}$. See Figure 5. We sample this field on a 100 by 100 grid, and calculate the persistence landscape of the sublevel set. For homology in degree 0, we truncate the infinite interval at the maximum value of the field. We calculate the mean persistence landscapes in degrees 0 and 1 from 100 samples (see Figure 5, where we have rescaled the filtration by a factor of 100).

In the Gaussian random field literature, it is more common to consider superlevel sets. However, by symmetry, the expected persistence landscape in this case is the same except for a change in the sign of the filtration.

We repeat this calculation for a similar Gaussian random field on $[0,1]^3$, this time using reduced homology. See Figure 5. This time we sample on a $25 \times 25 \times 25$ grid.

4.3 Torus and Sphere

Here we combine persistence landscapes and statistical inference to discriminate between iid samples of 1000 points from a torus and a sphere in $\mathbb{R}^3$ with the same surface area, using the uniform surface area measure as described by Diaconis et al. (2012) (see Figure 6). To be precise, we use the torus given by $(r-2)^2 + z^2 = 1$ in cylindrical coordinates, and the sphere given by $r^2 = 2\pi$ in spherical coordinates.

For these points, we construct a filtered simplicial complex as follows. First we triangulate the underlying space using the Coxeter–Freudenthal–Kuhn triangulation, starting with a cubical grid with sides of length $\frac{1}{2}$. Next we smooth our data using a triangular kernel with bandwidth 0.9. We evaluate this kernel density estimator at the vertices of our simplicial complex. Finally, we filter our simplicial complex as follows. For filtration level $-r$, we include a simplex in our triangulation if and only if the kernel density estimator has values greater than or equal to r at all of its vertices. Three stages in the filtration for one of the samples are shown in (see Figure 6). We then calculate the persistence landscape of this filtered simplicial complex for 100 samples and plot the mean landscapes (see Figure 6). We observe that the large peaks correspond to the Betti numbers of the torus and sphere.

Since the support of the persistence landscapes is bounded, we can use the integral of the landscapes to obtain a real valued random variable that satisfies (6). We use a two-sample z-test to test the null hypothesis that these random variables have equal mean. For the landscapes in dimensions 0 and 2 we cannot reject the null hypothesis. In dimension 1 we do reject the null hypothesis with a p-value of 3×10^{-6}.

We can also choose a functional that only integrates the persistence landscape $\lambda(k,t)$ for certain ranges of k. In dimension 1, with $k = 1$ or $k = 2$ there is a statistically significant difference (p-values of 10^{-8} and 3×10^{-6}), but not for $k > 2$. In dimension 2, there is not a significant difference for $k = 1$, but there is a significant difference for $k > 1$ (p-value $< 10^{-4}$).

Now we increase the difficulty by adding a fair amount of Gaussian noise to the point samples (see Figure 7) and using only 10 samples for each surface. This time we calculate the L^2 distances between the mean landscapes. We use the permutation test with 10,000 repetitions to determine if this distance is statistically significant. There is a significant difference in dimension 0, with a p value of 0.0111. This is surprising, since the mean landscapes look very similar. However, on closer inspection, they are shifted slightly (see Figure 7). Note that we are detecting a geometric difference, not a topological one. This shows that this statistic is quite powerful. There is also a significant difference in dimensions 1 and 2, with p values of 0.0000 and 0.0000, respectively.

5. Landscape Distance and Stability

In this section we define the landscape distance and use it to show that the persistence landscape is a stable summary statistic. We also show that the landscape distance gives lower bounds for the bottleneck and Wasserstein distances. We defer the proofs of the results of this section to the appendix.

Let M and M' be persistence modules as defined in Section 2.1 and let λ and λ' be their corresponding persistence landscapes as defined in Section 2.2. For $1 \leq p \leq \infty$, define the

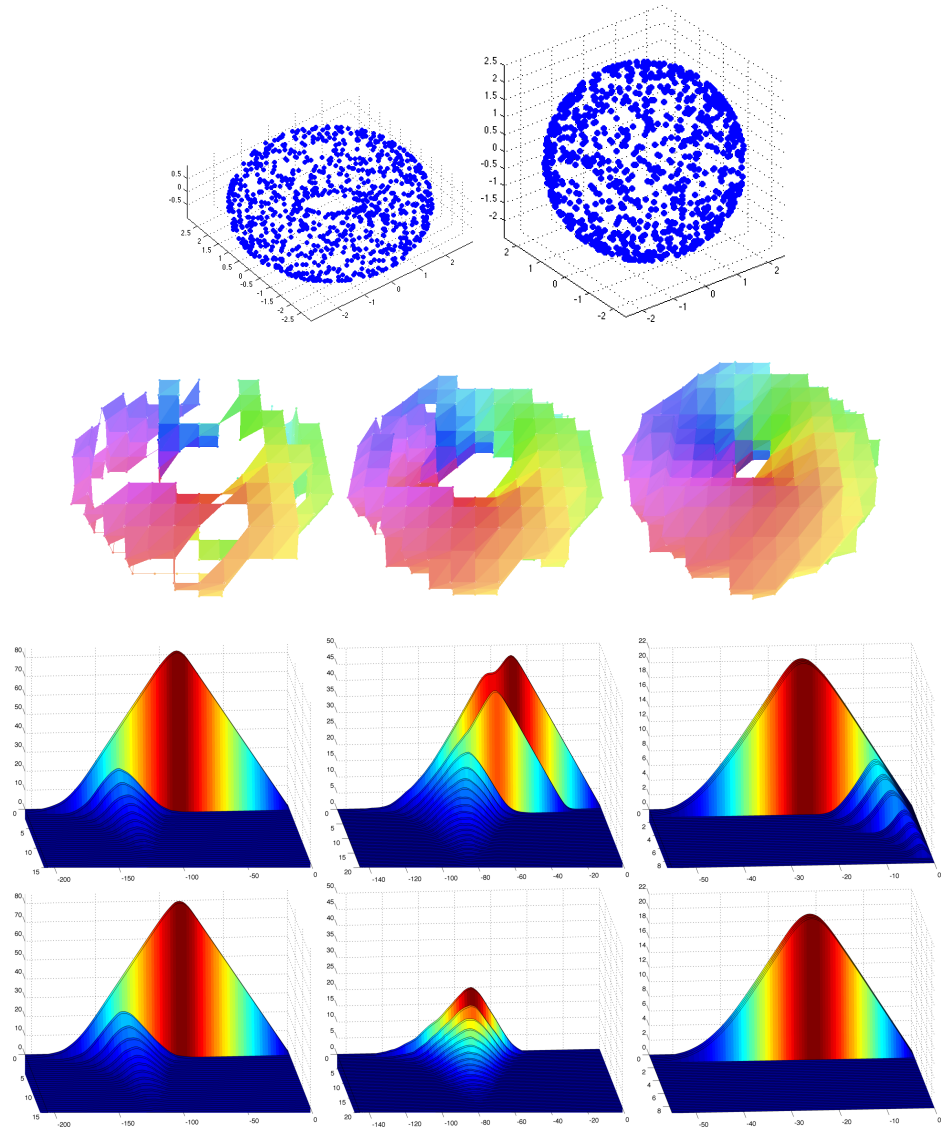


Figure 6: We sample 1000 points for a torus and sphere, 100 times each, construct the corresponding filtered simplicial complexes and calculate persistent homology. In columns 1, 2 and 3, we have the mean persistence landscape in dimension 0, 1 and 2 of the torus in row 3 and the sphere in row 4.

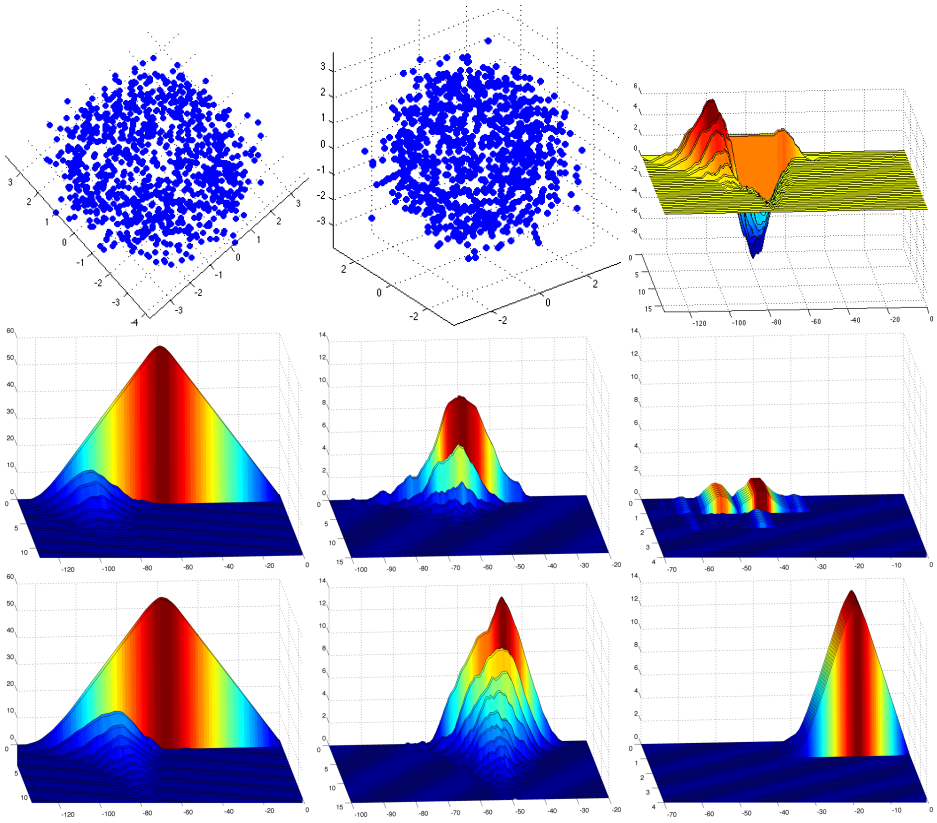


Figure 7: We again sample 1000 points sampled from a torus (top left) and sphere (top middle), this time with Gaussian noise. We show the torus from the perspective that makes it easiest to see the hole in the middle. We calculate persistent homology from 10 samples. In columns 1, 2 and 3, we have the mean persistence landscape in dimension 0, 1 and 2, respectively, with the torus in row 2 and the sphere in row 3. The top right is a graph of the difference between the mean landscapes in dimension 0.

p-landscape distance between M and M' by

$$\Lambda_p(M, M') = \|\lambda - \lambda'\|_p.$$

Similarly, if λ and λ' are the persistence landscapes corresponding to persistence diagrams D and D' (Section 2.3), then we define

$$\Lambda_p(D, D') = \|\lambda - \lambda\|_p.$$

Given a real valued function $f : X \to \mathbb{R}$ on a topological space X, let $M(f)$ denote be the corresponding persistence module defined at the end of Section 2.1.

Theorem 12 (∞-Landscape Stability Theorem) *Let $f, g : X \to \mathbb{R}$. Then*

$$\Lambda_\infty(M(f), M(g)) \leq \|f - g\|_\infty.$$

Thus the persistence landscape is stable with respect to the supremum norm. We remark that there are no assumptions on f and g, not even the q-tame condition of Chazal et al. (2012).

Let D be a persistence diagram. For $x = (b, d) \in D$, let $\ell = d - b$ denote the *persistence* of x. If $D = \{x_j\}$, let $\mathrm{Pers}_k(D) = \sum_j \ell_j^k$ denote the *degree-k total persistence* of D.

Now let us consider a persistence diagram to be an equivalence class of multisets of pairs (b, d) with $b \leq d$, where $D \sim D \amalg \{(t, t)\}$ for any $t \in \mathbb{R}$. That is, to any persistence diagram, we can freely adjoin points on the diagonal. This is reasonable, since points on the diagonal have zero persistence. Each persistence diagram has a unique representative $\hat{D}$ without any points on the diagonal. We set $|D| = |\hat{D}|$. We also remark that $\mathrm{Pers}_k(D)$ is well defined.

By allowing ourselves to add as many points on the diagonal as necessary, there exists bijections between any two persistence diagrams. Any bijection $\varphi : D \xrightarrow{\cong} D'$ can be represented by $\varphi : x_j \mapsto x'_j$, where $j \in J$ with $|J| = |D| + |D'|$. For a given φ, let $x_j = (b_j, d_j)$, $x'_j = (b'_j, d'_j)$ and $\varepsilon_j = \|x_j - x'_j\|_\infty = \max(|b_j - b'_j|, |d_j - d'_j|)$.

The *bottleneck distance* (Cohen-Steiner et al., 2007) between persistence diagrams D and D' is given by

$$W_\infty(D, D') = \inf_{\varphi : D \xrightarrow{\cong} D'} \sup_j \varepsilon_j,$$

where the infimum is taken over all bijections from D to D'. It follows that for the empty persistence diagram $\emptyset$, $W_\infty(D, \emptyset) = \frac{1}{2} \sup_j \ell_j$.

The ∞-landscape distance is bounded by the bottleneck distance.

Theorem 13 *For persistence diagrams D and D',*

$$\Lambda_\infty(D, D') \leq W_\infty(D, D').$$

For $p \geq 1$, the *p-Wasserstein distance* (Cohen-Steiner et al., 2010) between D and D' is given by

$$W_p(D, D') = \inf_{\varphi : D \xrightarrow{\cong} D'} \left[\sum_j \varepsilon_j^p \right]^{\frac{1}{p}}.$$

We remark that the Wasserstein distance gives equal weighting to the ε_j while the landscape distance gives a stronger weighting to ε_j if x_j has larger persistence. The landscape distance is most closely related to a weighted version of the Wasserstein distance that we now define. The *persistence weighted p-Wasserstein distance* between D and D' is given by

$$\overline{W}_p(D, D') = \inf_{\varphi: D \xrightarrow{\cong} D'} \left[\sum_j \ell_j \varepsilon_j^p \right]^{\frac{1}{p}}.$$

Note that it is asymmetric.

For the remainder of the section we assume that D and D' are finite. The following result bounds the p-landscape distance. Recall that ℓ_j is the persistence of $x_j \in D$ and when $\varphi : x_j \mapsto x'_j$, $\varepsilon_j = \|x_j - x'_j\|_\infty$

Theorem 14 *If $n = |D| + |D|$ then*

$$\Lambda_p(D, D')^p \leq \min_{\varphi: D \xrightarrow{\cong} D'} \left[\sum_{j=1}^{n} \ell_j \varepsilon_j^p + \frac{2}{p+1} \sum_{j=1}^{n} \varepsilon_j^{p+1} \right].$$

From this we can obtain a lower bound on the p-Wasserstein distance.

Corollary 15 $W_p(D, D')^p \geq \min\left(1, \frac{1}{2}\left[W_\infty(D, \emptyset) + \frac{1}{p+1}\right]^{-1} \Lambda_p(D, D')^p\right)$.

For our final stability theorem, we use ideas from Cohen-Steiner et al. (2010). Let $f : X \to \mathbb{R}$ be a function on a topological space. We say that f is *tame* if for all but finitely many $a \in \mathbb{R}$, the associated persistence module $M(f)$ is constant and finite dimensional on some open interval containing a. For such an f, let $D(f)$ denote the corresponding persistence diagram. If X is a metric space we say that f is *Lipschitz* if there is some constant c such that $|f(x) - f(y)| \leq c\, d(x, y)$ for all $x, y \in X$. We let $\mathrm{Lip}(f)$ denote the infimum of all such c. We say that a metric space X *implies bounded degree-k total persistence* if there is a constant $C_{X,k}$ such that $\mathrm{Pers}_k(D(f)) \leq C_{X,k}$ for all tame Lipschitz functions $f : X \to \mathbb{R}$ such that $\mathrm{Lip}(f) \leq 1$. For example, as observed by Cohen-Steiner et al. (2010), if X is the n-dimensional sphere, then $X = S^n$ has bounded k-persistence for $k = n + \delta$ for any $\delta > 0$, but does not have bounded k-persistence for $k < n$.

Theorem 16 (p-Landscape stability theorem) *Let X be a triangulable, compact metric space that implies bounded degree-k total persistence for some real number $k \geq 1$, and let f and g be two tame Lipschitz functions. Then*

$$\Lambda_p(D(f), D(g))^p \leq C \|f - g\|_\infty^{p-k},$$

for all $p \geq k$, where $C = C_{X,k}\|f\|_\infty(\mathrm{Lip}(f)^k + \mathrm{Lip}(g)^k) + C_{X,k+1}\frac{1}{p+1}(\mathrm{Lip}(f)^{k+1} + \mathrm{Lip}(g)^{k+1})$.

Thus the persistence diagram is stable with respect to the p-landscape distance if $p > k$, where X has bounded degree-k total persistence. This is the same condition as for the stability of the p-Wasserstein distance in Cohen-Steiner et al. (2010). Equivalently, the

persistence landscape is stable with respect to the p-norm if $p > k$, where X has bounded degree-k total persistence.

Acknowledgments

The author would like to thank Robert Adler, Frederic Chazal, Herbert Edelsbrunner, Giseon Heo, Sayan Mukherjee and Stephen Rush for helpful discussions. Thanks to Junyong Park for suggesting Hotelling's T^2 test. Also thanks to the anonymous referees who made a number of helpful comments to improve the exposition. In addition, the author gratefully acknowledges the support of the Air Force Office of Scientific Research (AFOSR grant FA9550-13-1-0115).

Appendix A. Proofs

Proof [Proof of Lemma 4(3)] We will prove that λ_k is 1-Lipschitz. That is, $|\lambda_k(t)-\lambda_k(s)| \leq |t-s|$, for all $s, t \in \mathbb{R}$.

Let $s, t \in \mathbb{R}$. Without loss of generality, assume that $\lambda_k(t) \geq \lambda_k(s) \geq 0$. If $\lambda_k(t) \leq |t-s|$, then $\lambda_k(t) - \lambda_k(s) \leq \lambda_k(t) \leq |t-s|$ and we are done. So assume that $\lambda_k(t) > |t-s|$.

Let $0 < h < \lambda_k(t) - |t-s|$. Then $t - \lambda_k(t) < s - h < s + h < t + \lambda_k(t)$. Thus, by Lemma 1 and Definition 3, $\beta^{s-h,s+h} \geq k$. It follows that $\lambda_k(s) \geq \lambda_k(t) - |t-s|$. Thus $\lambda_k(t) - \lambda_k(s) \leq |t-s|$. ■

Theorems 12 and 13 follow from the next result which is of independent interest. Following Chazal et al. (2009), we say that two persistence modules M and M' are *ε-interleaved* if for all $a \in \mathbb{R}$ there exist linear maps $\varphi_a : M_a \to M'_{a+\varepsilon}$ and $\psi : M'_a \to M_{a+\varepsilon}$ such that for all $a \in \mathbb{R}$, $\psi_{a+\varepsilon} \circ \varphi_a = M(a \leq a + 2\varepsilon)$ and $\varphi_{a+\varepsilon} \circ \psi_a = M'(a \leq a + 2\varepsilon)$ and for all $a \leq b$ $M'(a + \varepsilon \leq b + \varepsilon) \circ \varphi_a = \varphi_b \circ M(a \leq b)$ and $M(a + \varepsilon \leq b + \varepsilon) \circ \psi_a = \psi_b \circ M'(a \leq b)$. For persistence modules M and M' define the *interleaving distance* between M and M' by

$$d_I(M, M) = \inf(\varepsilon \mid M \text{ and } M' \text{ are } \varepsilon\text{-interleaved}).$$

Theorem 17 $\Lambda_\infty(M, M') \leq d_I(M, M')$.

Proof Assume that M and M' are ε-interleaved. Then for $t \in \mathbb{R}$ and $m \geq \varepsilon$, the map $M(t - m \leq t + m)$ factors through the map $M'(t - m + \varepsilon \leq t + m - \varepsilon)$. So by Lemma 1, $\beta^{t-m+\varepsilon,t+m-\varepsilon}(M') \geq \beta^{t-m,t+m}(M)$. Thus by Definition 3, $\lambda'(k, t) \geq \lambda(k, t) - \varepsilon$ for all $k \geq 1$. It follows that $\|\lambda - \lambda'\|_\infty \leq \varepsilon$. ■

Proof [Proof of Theorem 12] Combining Theorem 17 with the stability theorem of Bubenik and Scott (2014), we have $\Lambda_\infty(M(f), M(g)) \leq d_I(M(f), M(g)) \leq \|f - g\|_\infty$. ■

Proof [Proof of Theorem 13] For a persistence diagram D, consider the persistence module given by the corresponding sum of interval modules (Chazal et al., 2012), $M(D) = \oplus_{(a,b)\in\hat{D}} \mathbb{I}(a, b)$. Combining Theorem 17 with Theorem 4.9 of Chazal et al. (2012) we have

$\Lambda_\infty(M(D), M(D')) \leq d_I(M(D), M(D')) \leq W_\infty(D, D')$. ■

Proof [Proof of Theorem 14] Let $\varphi : D \xrightarrow{\cong} D'$ with $\varphi(x_j) = x'_j$. Let $\lambda = \lambda(D)$ and $\lambda' = \lambda(D')$. So $\Lambda_p(D, D')^p = \|\lambda - \lambda'\|_p^p$.

$$\begin{aligned}\|\lambda - \lambda'\|_p^p &= \int |\lambda(k,t) - \lambda'(k,t)|^p \\ &= \sum_{k=1}^n \int |\lambda_k(t) - \lambda'_k(t)|^p \, dt \\ &= \int \sum_{k=1}^n |\lambda_k(t) - \lambda'_k(t)|^p \, dt\end{aligned}$$

Fix t. Let $u_j(t) = \lambda(\{x_j\})(1,t)$ and $v_j(t) = \lambda(\{x'_j\})(1,t)$. For each t, let $u_{(1)}(t) \leq \cdots \leq u_{(n)}(t)$ denote an ordering of $u_1(t), \ldots, u_n(t)$ and define $v_{(k)}(t)$ for $1 \leq k \leq n$ similarly. Then $u_{(k)}(t) = \lambda_k(t)$ and $v_{(k)}(t) = \lambda'_k(t)$ (see Figure 2). We obtain the result from the following where the two inequalities are proven in Lemmata 18 and 19.

$$\begin{aligned}\|\lambda - \lambda'\|_p^p &= \int \sum_{k=1}^n |u_{(k)}(t) - v_{(k)}(t)|^p \, dt \\ &\leq \int \sum_{k=1}^n |u_k(t) - v_k(t)|^p \, dt \\ &= \sum_{j=1}^n \int |u_j(t) - v_j(t)|^p \, dt \\ &\leq \sum_{j=1}^n \ell_j \varepsilon_j^p + \frac{2}{p+1} \sum_{j=1}^n \varepsilon_j^{p+1}.\end{aligned}$$

■

Lemma 18 *Let $u_1, \ldots, u_n \in \mathbb{R}$ and $v_1, \ldots, v_n \in \mathbb{R}$. Order them $u_{(1)} \leq \cdots \leq u_{(n)}$ and $v_{(1)} \leq \cdots \leq v_{(n)}$. Then*

$$\sum_{j=1}^n |u_{(j)} - v_{(j)}|^p \leq \sum_{j=1}^n |u_j - v_j|^p.$$

Proof Assume $u_1 < \cdots < u_n$, $v_1 < \cdots < v_n$, and $p \geq 1$. Let u and v denote $(u_1, \ldots, u_n)$ and $(v_1, \ldots, v_n)$. Let Σ_n denote the symmetric group on n letters and let $f_n : \Sigma_n \to \mathbb{R}$ be given by $f_n(\sigma) = \sum_{j=1}^n |u_j - v_{\sigma(j)}|^p$. We will prove by induction that if $f_n(\sigma)$ is minimal then σ is the identity, which we denote by 1.

For $n = 1$ this is trivial. For $n = 2$ assume without loss of generality that $u_1 = 0$, $u_2 = 1$ and $0 \leq v_1 < v_2$. Let 1 and τ denote the elements of Σ_2. Then $f(1) = v_1^p + |1 - v_2|^p$ and $f(\tau) = v_2^p + |1 - v_1|^p$. Notice that $f(1) < f(\tau)$ if and only if $v_1^p - |1 - v_1|^p < v_2^p - |1 - v_2|^p$. The result follows from checking that $g(x) = x^p - |1 - x|^p$ is an increasing function for $x \geq 0$.

Now assume that the statement is true for some $n \geq 2$. Assume that $f_{n+1}(\sigma^*)$ is minimal. Fix $1 \leq i \leq n+1$. Let $u' = (u_1, \ldots, \hat{u}_i, \ldots, u_{n+1})$ and $v' = (v_1, \ldots, \hat{v}_{\sigma^*(i)}, \ldots, v_{n+1})$, where $\hat{\cdot}$ denotes omission. Since $f_{n+1}(\sigma^*)$ is minimal for u and v, it follows that $\sum_{j=1, j\neq i}^{n} |u_j - v_{\sigma^*(j)}|$ is minimal for u' and v'. By the induction hypothesis, for $1 \leq j < k \leq n+1$ and $j, k \neq i$, $\sigma^*(j) < \sigma^*(k)$. Therefore $\sigma^* = 1$. Thus, by induction, the statement is true for all n.

Hence $\sum_{j=1}^{n} |u_{(j)} - v_{(j)}|^p \leq \sum_{j=1}^{n} |u_j - v_j|^p$ if $u_{(1)} < \cdots < u_{(n)}$ and $v_{(1)} < \cdots < v_{(n)}$. The statement in the lemma follows by continuity. ■

Lemma 19 *Let $x = (b, d)$ and $x' = (b', d')$ where $b \leq d$ and $b' \leq d'$. Let $\ell = d - b$ and $\varepsilon = \|x - x'\|_\infty$. Then $\|\lambda(\{x\}) - \lambda(\{x'\})\|_p^p \leq \ell\varepsilon^p + \frac{2}{p+1}\varepsilon^{p+1}$.*

Proof Let $\lambda = \lambda(\{x\})$ and $\lambda' = \lambda(\{x'\})$. First $\lambda_k = \lambda'_k = 0$ for $k > 1$; so $\|\lambda - \lambda'\|_p = \|\lambda_1 - \lambda'_1\|_p$. Second $\lambda_1(t) = (h - |t - m|)_+$, where $h = \frac{d-b}{2}$, $m = \frac{b+d}{2}$, and $y_+ = \max(y, 0)$, and similarly for λ'_1 (see Figure 2).

Fix x and ε. As x' moves along the square $\|x - x'\|_\infty = \varepsilon$, $\|\lambda_1 - \lambda'_1\|_p^p$ has a maximum if $x' = (a - \varepsilon, b + \varepsilon)$. In this case $\|\lambda_1 - \lambda'_1\|_p^p = 2\int_0^h \varepsilon^p \, dt + 2\int_0^\varepsilon t^p \, dt = \ell\varepsilon^p + \frac{2}{p+1}\varepsilon^{p+1}$. ■

Proof [Proof of Corollary 15] Let $\varphi : D \xrightarrow{\cong} D'$ be a minimizer for $W_p(D, D')$, with corresponding $\{\varepsilon_j\}$. Assume that $W_p(D, D') \leq 1$. Then $W_p(D, D')^p = \sum_{j=1}^{n} \varepsilon_j^p \leq 1$. So for $1 \leq j \leq n$, $\varepsilon_j \leq 1$. Combining this with Theorem 14, we have that

$$\Lambda_p(D, D')^p \leq \sum_{j=1}^{n} \left(\ell_j + \frac{2}{p+1} \right) \varepsilon_j^p. \tag{8}$$

Since $W_\infty(D, \emptyset) = \max \frac{1}{2}\ell_j$, $\ell_j \leq 2\, W_\infty(D, \emptyset)$. Hence

$$\Lambda_p(D, D')^p \leq 2 \left(W_\infty(D, \emptyset) + \frac{1}{p+1} \right) W_p(D, D')^p. \tag{9}$$

Therefore $W_p(D, D')^p \geq 1$ or $W_p(D, D')^p \geq \frac{1}{2}\left[W_\infty(D, \emptyset) + \frac{1}{p+1} \right]^{-1} \Lambda_p(D, D')^p$. The statement of the corollary follows. ■

Theorem 16 follows from the following corollary to Theorem 14 which is of independent interest.

Corollary 20 *Let $p \geq k \geq 1$. Then*

$$\Lambda_p(D, D')^p \leq W_\infty(D, D')^{p-k} \Big[W_\infty(D, \emptyset)(\mathrm{Pers}_k(D) + \mathrm{Pers}_k(D')) + \frac{1}{p+1}(\mathrm{Pers}_{k+1}(D) + \mathrm{Pers}_{k+1}(D')) \Big]$$

Proof Let φ be a minimizer for $W_\infty(D, D')$ with corresponding $\{\varepsilon_j\}$. If $\varepsilon_j > \frac{\ell_j}{2} + \frac{\ell'_j}{2}$ then modify φ to pair $x_j = (b_j, d_j)$ with $\bar{x}_j = (\frac{b_j+d_j}{2}, \frac{b_j+d_j}{2})$ and similarly for x'_j. Note that $\|x_j - \bar{x}_j\|_\infty = \frac{\ell_j}{2}$ and $\|x'_j - \bar{x'}_j\|_\infty = \frac{\ell'_j}{2}$, so φ is still a minimizer for $W_\infty(D, D')$.

Recall that for all j, $\ell_j \leq 2\, W_\infty(D, \emptyset)$. Since φ is a minimizer for $W_\infty(D, D')$, for all j, $\varepsilon_j \leq W_\infty(D, D')$. So applying our choice of φ to Theorem 14 we have,

$$\Lambda_p(D, D')^p \leq W_\infty(D, D')^{p-k} \left[2\, W_\infty(D, \emptyset) \sum_{j=1}^{n} \varepsilon_j^k + \frac{2}{p+1} \sum_{j=1}^{n} \varepsilon_j^{k+1} \right].$$

Now $\varepsilon_j^q \leq \left(\frac{\ell_j}{2} + \frac{\ell'_j}{2}\right)^q \leq \frac{1}{2}\left((\ell_j)^q + (\ell'_j)^q\right)$ for $q \geq 1$, where the right hand side follows by the convexity of $\alpha(x) = x^q$ for $q \geq 1$. Thus $\sum_{j=1}^n \varepsilon_j^q \leq \frac{1}{2}(\mathrm{Pers}_q(D) + \mathrm{Pers}_q(D'))$ for $q \geq 1$. The result follows. ■

Proof [Proof of Theorem 16] Theorem 16 follows from Corollary 20 by the following two observations. First, by the stability theorem of Cohen-Steiner et al. (2007), $W_\infty(D(f), D(g)) \leq \|f - g\|_\infty$ and $W_\infty(D(f), \emptyset) \leq \|f\|_\infty$. Second, if $\mathrm{Pers}_q(D(f)) \leq C_{X,q}$ for all tame Lipschitz functions $f : X \to \mathbb{R}$ with $\mathrm{Lip}(f) \leq 1$, then for general tame Lipschitz functions, $\mathrm{Pers}_q(D(f)) \leq C_{X,q} \mathrm{Lip}(f)^q$. ■

References

Robert J. Adler and Jonathan E. Taylor. *Random Fields and Geometry*. Springer Monographs in Mathematics. Springer, New York, 2007. ISBN 978-0-387-48112-8.

Robert J. Adler, Omer Bobrowski, Matthew S. Borman, Eliran Subag, and Shmuel Weinberger. Persistent homology for random fields and complexes. In *Borrowing Strength: Theory Powering Applications—a Festschrift for Lawrence D. Brown*, volume 6 of *Inst. Math. Stat. Collect.*, pages 124–143. Inst. Math. Statist., Beachwood, OH, 2010.

Andrew J. Blumberg, Itamar Gal, Michael A. Mandell, and Matthew Pancia. Robust statistics, hypothesis testing, and confidence intervals for persistent homology on metric measure spaces. *Found. Comput. Math.*, 14(4):745–789, 2014. ISSN 1615-3375. doi: 10.1007/s10208-014-9201-4. URL `http://dx.doi.org/10.1007/s10208-014-9201-4`.

Omer Bobrowski and Matthew Strom Borman. Euler integration of Gaussian random fields and persistent homology. *J. Topol. Anal.*, 4(1):49–70, 2012. ISSN 1793-5253.

Karol Borsuk. On the imbedding of systems of compacta in simplicial complexes. *Fund. Math.*, 35:217–234, 1948. ISSN 0016-2736.

Peter Bubenik and Jonathan A. Scott. Categorification of persistent homology. *Discrete Comput. Geom.*, 51(3):600–627, 2014. ISSN 0179-5376.

Peter Bubenik, Gunnar Carlsson, Peter T. Kim, and Zhi-Ming Luo. Statistical topology via Morse theory persistence and nonparametric estimation. In *Algebraic Methods in Statistics and Probability II*, volume 516 of *Contemp. Math.*, pages 75–92. Amer. Math. Soc., Providence, RI, 2010.

Gunnar Carlsson. Topology and data. *Bull. Amer. Math. Soc. (N.S.)*, 46(2):255–308, 2009. ISSN 0273-0979.

Gunnar Carlsson, Tigran Ishkhanov, Vin de Silva, and Afra Zomorodian. On the local behavior of spaces of natural images. *Int. J. Comput. Vision*, 76(1):1–12, 2008. ISSN 0920-5691.

Frédéric Chazal, David Cohen-Steiner, Marc Glisse, Leonidas J. Guibas, and Steve Y. Oudot. Proximity of persistence modules and their diagrams. In *Proceedings of the 25th Annual Symposium on Computational Geometry*, SCG '09, pages 237–246, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-501-7.

Frederic Chazal, Vin de Silva, Marc Glisse, and Steve Oudot. The structure and stability of persistence modules. arXiv:1207.3674 [math.AT], 2012.

Frédéric Chazal, Marc Glisse, Catherine Labruère, and Bertrand Michel. Optimal rates of convergence for persistence diagrams in topological data analysis. 2013. arXiv:1305.6239 [math.ST].

Frédéric Chazal, Brittany Terese Fasy, Fabrizio Lecci, Alessandro Rinaldo, and Larry Wasserman. Stochastic convergence of persistence landscapes and silhouettes. *Symposium on Computational Geometry (SoCG)*, 2014.

Chao Chen and Michael Kerber. An output-sensitive algorithm for persistent homology. *Comput. Geom.*, 46(4):435–447, 2013. ISSN 0925-7721.

Moo K. Chung, Peter Bubenik, and Peter T. Kim. Persistence diagrams in cortical surface data. In *Information Processing in Medical Imaging (IPMI) 2009*, volume 5636 of *Lecture Notes in Computer Science*, pages 386–397, 2009.

David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete Comput. Geom.*, 37(1):103–120, 2007. ISSN 0179-5376.

David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Extending persistence using Poincaré and Lefschetz duality. *Found. Comput. Math.*, 9(1):79–103, 2009. ISSN 1615-3375.

David Cohen-Steiner, Herbert Edelsbrunner, John Harer, and Yuriy Mileyko. Lipschitz functions have L_p-stable persistence. *Found. Comput. Math.*, 10(2):127–139, 2010. ISSN 1615-3375.

William Crawley-Boevey. Decomposition of pointwise finite-dimensional persistence modules. arXiv:1210.0819 [math.RT], 2012.

Vin de Silva and Gunnar Carlsson. Topological estimation using witness complexes. *Eurographics Symposium on Point-Based Graphics*, 2004.

Vin De Silva and Robert Ghrist. Coverage in sensor networks via persistent homology. *Algebr. Geom. Topol.*, 7:339–358, 2007a.

Vin De Silva and Robert Ghrist. Homological sensor networks. *Notic. Amer. Math. Soc.*, 54(1):10–17, 2007b.

Mary-Lee Dequéant, Sebastian Ahnert, Herbert Edelsbrunner, Thomas M. A. Fink, Earl F. Glynn, Gaye Hattem, Andrzej Kudlicki, Yuriy Mileyko, Jason Morton, Arcady R. Mushegian, Lior Pachter, Maga Rowicka, Anne Shiu, Bernd Sturmfels, and Olivier Pourquié. Comparison of pattern detection methods in microarray time series of the segmentation clock. *PLoS ONE*, 3(8):e2856, 08 2008.

Tamal Krishna Dey, Fengtao Fan, and Yusu Wang. Graph induced complex on point data. In *Proceedings of the Twenty-ninth Annual Symposium on Computational Geometry*, SoCG '13, pages 107–116, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2031-3.

Persi Diaconis, Susan Holmes, and Mehrdad Shahshahani. Sampling from a manifold. arXiv:1206.6913 [math.ST], 2012.

Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. *Discrete Comput. Geom.*, 28(4):511–533, 2002. ISSN 0179-5376. Discrete and computational geometry and graph drawing (Columbia, SC, 2001).

Herbert Edelsbrunner, Dmitriy Morozov, and Amit Patel. Quantifying transversality by measuring the robustness of intersections. *Found. Comput. Math.*, 11(3):345–361, 2011. ISSN 1615-3375.

Brittany Terese Fasy, Fabrizio Lecci, Alessandro Rinaldo, Larry Wasserman, Sivaraman Balakrishnan, and Aarti Singh. Confidence sets for persistence diagrams. *Ann. Statist.*, 42(6):2301–2339, 2014. ISSN 0090-5364. doi: 10.1214/14-AOS1252. URL `http://dx.doi.org/10.1214/14-AOS1252`.

Robert Ghrist. Barcodes: the persistent topology of data. *Bull. Amer. Math. Soc. (N.S.)*, 45(1):61–75, 2008. ISSN 0273-0979.

Allen Hatcher. *Algebraic Topology*. Cambridge University Press, Cambridge, 2002. ISBN 0-521-79160-X; 0-521-79540-0.

Giseon Heo, Jennifer Gamble, and Peter T. Kim. Topological analysis of variance and the maxillary complex. *J. Amer. Statist. Assoc.*, 107(498):477–492, 2012. ISSN 0162-1459.

J. Hoffmann-Jørgensen and G. Pisier. The law of large numbers and the central limit theorem in Banach spaces. *Ann. Probability*, 4(4):587–599, 1976.

Violeta Kovacev-Nikolic, Giseon Heo, Dragan Nikolić, and Peter Bubenik. Using cycles in high dimensional data to analyze protein binding. 2014. arXiv:1412.1394 [stat.ME].

Michel Ledoux and Michel Talagrand. *Probability in Banach Spaces*. Classics in Mathematics. Springer-Verlag, Berlin, 2011. ISBN 978-3-642-20211-7. Isoperimetry and processes, Reprint of the 1991 edition.

Yuriy Mileyko, Sayan Mukherjee, and John Harer. Probability measures on the space of persistence diagrams. *Inverse Problems*, 27(12):124007, 22, 2011. ISSN 0266-5611.

Nikola Milosavljević, Dmitriy Morozov, and Primož Škraba. Zigzag persistent homology in matrix multiplication time. In *Computational Geometry (SCG'11)*, pages 216–225. ACM, New York, 2011.

Dimitriy Morozov. Dionysus: a C++ library with various algorithms for computing persistent homology. Software available at `http://www.mrzv.org/software/dionysus/`, 2012.

Elizabeth Munch, Paul Bendich, Katharine Turner, Sayan Mukherjee, Jonathan Mattingly, and John Harer. Probabilistic fréchet means and statistics on vineyards. 2013. arXiv:1307.6530 [math.PR].

Vidit Nanda. Perseus: the persistent homology software. Software available at `http://www.math.rutgers.edu/~vidit/perseus/index.html`, 2013.

Monica Nicolau, Arnold J. Levine, and Gunnar Carlsson. Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival. *Proc. Nat. Acad. Sci.*, 108(17):7265–7270, 2011.

Andrew Robinson and Katharine Turner. Hypothesis testing for topological data analysis. 2013. arXiv:1310.7467 [stat.AP].

Andrew Tausz, Mikael Vejdemo-Johansson, and Henry Adams. Javaplex: a research software package for persistent (co)homology. Software available at `http://code.google.com/javaplex`, 2011.

Katharine Turner, Yuriy Mileyko, Sayan Mukherjee, and John Harer. Fréchet means for distributions of persistence diagrams. *Discrete Comput. Geom.*, 52(1):44–70, 2014.

Andrew T. A. Wood and Grace Chan. Simulation of stationary Gaussian processes in $[0, 1]^d$. *J. Comput. Graph. Statist.*, 3(4):409–432, 1994. ISSN 1061-8600.

Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. *Discrete Comput. Geom.*, 33(2):249–274, 2005. ISSN 0179-5376.

Journal of Machine Learning Research 16 (2015) 103-147 Submitted 6/11; Revised 7/14; Published 1/15

Links Between Multiplicity Automata, Observable Operator Models and Predictive State Representations — a Unified Learning Framework

Michael Thon M.THON@JACOBS-UNIVERSITY.DE
Herbert Jaeger H.JAEGER@JACOBS-UNIVERSITY.DE
Jacobs University Bremen
28759 Bremen, Germany

Editor: Joelle Pineau

Abstract

Stochastic multiplicity automata (SMA) are weighted finite automata that generalize probabilistic automata. They have been used in the context of probabilistic grammatical inference. Observable operator models (OOMs) are a generalization of hidden Markov models, which in turn are models for discrete-valued stochastic processes and are used ubiquitously in the context of speech recognition and bio-sequence modeling. Predictive state representations (PSRs) extend OOMs to stochastic input-output systems and are employed in the context of agent modeling and planning.

We present SMA, OOMs, and PSRs under the common framework of sequential systems, which are an algebraic characterization of multiplicity automata, and examine the precise relationships between them. Furthermore, we establish a unified approach to learning such models from data. Many of the learning algorithms that have been proposed can be understood as variations of this basic learning scheme, and several turn out to be closely related to each other, or even equivalent.

Keywords: multiplicity automata, hidden Markov models, observable operator models, predictive state representations, spectral learning algorithms

1. Introduction

Multiplicity automata (MA) (Schützenberger, 1961) are weighted nondeterministic automata which generalize both finite and probabilistic automata. The discovery that MA are efficiently learnable (Bergadano and Varricchio, 1994; Ohnishi et al., 1994) in the exact learning model of Angluin (Angluin, 1987) sparked an interest in these, and several versions have been studied. One such version is stochastic multiplicity automata (SMA), which model rational stochastic languages and have been used in the context of probabilistic grammatical inference (Denis et al., 2006; Bailly et al., 2009). Independent of this line of research, hidden Markov models (HMMs) (see Rabiner, 1989) for discrete-valued stochastic processes have been extensively studied and are now a standard tool in many pattern recognition domains such as speech recognition, natural language processing and bio-sequence modeling. Observable operator models (OOMs) are a generalization of HMMs that was introduced by Jaeger (1998) following previous work on deciding the equivalence of HMMs (Ito et al., 1992). Finally, predictive state representations (PSRs) are mod-

els for stochastic input-output systems developed by Littman, Sutton, and Singh (2001) and inspired by OOMs. PSRs generalize partially observable Markov decision processes (POMDPs) (Kaelbling et al., 1998) and have been used in the context of agent modeling and planning (James et al., 2004; James and Singh, 2005; Wolfe and Singh, 2006; Boots et al., 2010). As it turns out, all of these models are instances of MA and thereby closely related, though this is not widely perceived, due in part to the disjoint scientific communities.

All of SMA, OOMs and PSRs model some form of probability distribution. A central task common to all cases is therefore to estimate a model from a given sample. This may also be referred to as learning, system identification or model induction depending on the context.

In this paper we present SMA, OOMs, and PSRs under a common framework and examine the precise relationships between them. Furthermore, we establish a unified approach to learning such models from data. Many of the learning algorithms that have been proposed can be understood as variations of this basic learning theme, and several turn out to be closely related or even equivalent.

In Section 2 we cover the essential theory for sequential systems (SSs) — a term coined by Carlyle and Paz (1971) for a purely algebraic characterization of MA. Though not new, we present this theory in a way that can be readily turned into algorithms, and with full proofs, because they give much insight and pave the way to the presented learning approach. The first result concerns the relationship between SSs and the objects that they describe, namely *formal series* $f : \Sigma^* \to K$ for $K = \mathbb{R}$ or $K = \mathbb{C}$ (see Section 1.1 for details). Any such function can be associated with a linear function space $\mathcal{F}$, and has a SS representation if and only if the space $\mathcal{F}$ is finite dimensional. In fact, a SS can be seen as a representation of f w.r.t. some basis of $\mathcal{F}$, and a change of basis will correspond to an equivalence transformation of SSs, where equivalence of two SSs means that they represent the same function. The remaining theory will be concerned with such transformations of SSs. It is shown how to transform any SS to an equivalent minimal SS, how to decide equivalence, how to normalize SSs and how to convert SSs into a so-called "interpretable" form.

In Section 3 we mention the relationship between MA and the more general class of weighted finite automata (WFA) and their extension to input-output systems called weighted finite-state transducers (WFST). We then present SMA, OOMs and PSRs as instances of SSs with specific additional constraints that model probabilistic languages, stochastic processes and controlled processes, respectively, via the formal series f that they describe. We only sketch the basic concepts and give pointers to relevant literature. The main emphasis is on exploring the relations among the various model classes. We show that SMA are related to OOMs in the same way that probabilistic finite automata are related to HMMs, and show how to trivially convert any HMM into an OOM. OOMs and PSRs share the notion of a "predictive state" for the modeled process, which can be either implicit (as in the case of OOMs) or explicit (as for PSRs). Any PSR is essentially an input-output (IO)-OOM, while any OOM can be converted to a PSR by making the state "interpretable". Finally, PSRs generalize POMDPs in the same way that OOMs generalize HMMs.

Section 4 on learning is the main technical contribution of this paper. We present a learning framework that covers the cases of SMA, OOMs and PSRs in a unified way. The

only difference for the model classes concerns the way that estimates are obtained from the sample data. To turn the learning framework into a concrete algorithm, several design choices need to be made. Depending on these, many algorithms that have been proposed in the literature are recovered. This unified viewpoint has several advantages. First of all, modifications and improvements made for a specific model class can be generalized to other learning algorithms. Additionally, the general learning framework allows us to identify the key points responsible for statistical efficiency and thereby indicates a clear path for improvements. In this section, we present generalized and simplified versions of two key OOM learning algorithms — error controlling (EC) and efficiency sharpening (ES) — and show that these are in fact closely related to spectral learning algorithms.

1.1 Notation

Let Σ^* be the set of words over a finite alphabet Σ, including the empty word ε. Symbols from the alphabet Σ will be denoted by normal variables as in $x, y \in \Sigma$, while words will be denoted by variables with a bar over them, e.g., $\overline{x}, \overline{y} \in \Sigma^*$. For $\overline{x}$ and $\overline{y}$ in Σ^*, let $\overline{xy}$ be the concatenation of words, and $|\overline{x}|$ denote the length of the word $\overline{x}$. Furthermore, let Σ^k denote the subset of words of length k. Let $\{\overline{x}_i \,|\, i \in \mathbb{N}\} = \Sigma^*$ be an enumeration of Σ^* such that $\overline{x}_0 = \varepsilon$. We will be interested in characterizing functions $f : \Sigma^* \to K$ for $K = \mathbb{R}$ or $K = \mathbb{C}$, since these can be used to describe probabilistic languages, stochastic processes and controlled processes (cf. Definitions 18, 20, and 28). These form a K-vector space which we denote by $K\langle\langle\Sigma\rangle\rangle$. For a given function $f : \Sigma^* \to K$, we define the *system matrix* F as the infinite matrix $F = [f(\overline{x}_j\overline{y}_i)]_{i,j\in\mathbb{N}}$. Note that this is the transpose of what is commonly known as the *Hankel matrix*. Furthermore, for a given function f we define the functions $f_{\overline{x}} : \Sigma^* \to K$ by setting $f_{\overline{x}}(\overline{y}) := f(\overline{xy})$ for any sequences $\overline{x}, \overline{y} \in \Sigma^*$. Note that these functions correspond to the columns of the system matrix F. Let $\mathcal{F} := \operatorname{span}\{f_{\overline{x}} \,|\, \overline{x} \in \Sigma^*\}$ be the linear space spanned by these functions / the columns of F. Clearly, $\mathcal{F} \subseteq K\langle\langle\Sigma\rangle\rangle$. We define $\operatorname{rank}(f) := \operatorname{rank}(F) = \operatorname{rank}(\mathcal{F})$.

A *d-dimensional sequential system (SS)* is a structure $\mathcal{M} = (\sigma, \{\tau_z\}, \omega_\varepsilon)$, which consists of an *initial state vector* $\omega_\varepsilon \in K^d$, a matrix $\tau_z \in K^{d\times d}$ for each $z \in \Sigma$ and an *evaluation function* $\sigma : K^d \to K$. For $\overline{x} = x_1 \cdots x_n \in \Sigma^*$ let $\tau_{\overline{x}} = \tau_{x_n} \cdots \tau_{x_1}$, and let $\omega_{\overline{x}} = \tau_{\overline{x}}\omega_\varepsilon$, which we call a *state* of the SS $\mathcal{M}$. Let $\tau_\Sigma = \sum_{z\in\Sigma} \tau_z$.

If the function σ is linear, we call the sequential system *linear*. In this paper, we will be dealing only with the linear case, so σ will just be a row vector, i.e., $\sigma^\top \in K^d$.

For a given SS $\mathcal{M}$, we define its *(external) function* to be

$$f_\mathcal{M} : \Sigma^* \to K, \quad f_\mathcal{M}(\overline{x}) = \sigma\tau_{\overline{x}}\omega_\varepsilon \tag{1}$$

Finally, we define the *rank* of a SS $\mathcal{M}$ to be $\operatorname{rank}(\mathcal{M}) := \operatorname{rank}(f_\mathcal{M})$.

2. Basic Properties of Sequential Systems

In this section we present the basic theory for sequential systems. This goes back to Schützenberger (1961), to Carlyle and Paz (1971) who coined the term *sequential systems*, and to Fliess (1974) but has been presented in various forms also for OOMs (Jaeger, 2000b) and PSRs (Singh et al., 2004). Here, we present the theory in a concise, self-contained fashion that can readily be turned into algorithms.

We begin with a technical result that lies at the heart of the whole theory.

Proposition 1 *Let $f : \Sigma^* \to K$ be given. If* $\operatorname{rank}(f) = d < \infty$*, then there exist linear operators $\tilde{\tau}_z : \mathcal{F} \to \mathcal{F}$ for each $z \in \Sigma$ and a linear functional $\tilde{\sigma} : \mathcal{F} \to K$ that satisfy $\tilde{\tau}_z(f_{\overline{x}}) = f_{\overline{x}z}$ and $\tilde{\sigma}(f_{\overline{x}}) = f(\overline{x})$ for all $\overline{x} \in \Sigma^*$. Furthermore, $\tilde{\sigma}(\tilde{\tau}_{\overline{x}}(f_\varepsilon)) = f(\overline{x})$ for all $\overline{x} \in \Sigma^*$, where $\tilde{\tau}_{\overline{x}} = \tilde{\tau}_{x_n} \circ \cdots \circ \tilde{\tau}_{x_1}$.*

Proof Let $J \subset \mathbb{N}$ be an index set denoting a maximal set of linearly independent columns of the matrix F. Then clearly, $\mathcal{B} = \{f_{\overline{x}_j} \mid j \in J\}$ is a basis for $\mathcal{F}$. Define linear operators $\tilde{\tau}_z$ and a linear functional $\tilde{\sigma}$ by their action on these basis elements:

- $\tilde{\tau}_z(f_{\overline{x}_j}) := f_{\overline{x}_j z}$ for all $z \in \Sigma$,
- $\tilde{\sigma}(f_{\overline{x}_j}) := f_{\overline{x}_j}(\varepsilon) = f(\overline{x}_j)$.

We will show that then $\tilde{\tau}_z(f_{\overline{x}}) = f_{\overline{x}z}$ and $\tilde{\sigma}(f_{\overline{x}}) = f(\overline{x})$ for all $\overline{x} \in \Sigma^*$. For this, let $\overline{x} \in \Sigma^*$. Then $f_{\overline{x}} = \sum_{j\in J} \lambda_j f_{\overline{x}_j}$ for suitable coordinates λ_j, and $f_{\overline{x}z} = \sum_{j\in J} \lambda_j f_{\overline{x}_j z}$, since for any $\overline{y} \in \Sigma^*$, we have $f_{\overline{x}z}(\overline{y}) = f_{\overline{x}}(z\overline{y}) = \sum_{j\in J} \lambda_j f_{\overline{x}_j}(z\overline{y}) = \sum_{j\in J} \lambda_j f_{\overline{x}_j z}(\overline{y})$. Therefore, $\tilde{\tau}_z(f_{\overline{x}}) = \tilde{\tau}_z(\sum_{j\in J} \lambda_j f_{\overline{x}_j}) = \sum_{j\in J} \lambda_j f_{\overline{x}_j z} = f_{\overline{x}z}$, and $\tilde{\sigma}(f_{\overline{x}}) = \tilde{\sigma}(\sum_{j\in J} \lambda_j f_{\overline{x}_j}) = \sum_{j\in J} \lambda_j f_{\overline{x}_j}(\varepsilon) = f_{\overline{x}}(\varepsilon) = f(\overline{x})$.

Finally, $\tilde{\sigma}(\tilde{\tau}_{\overline{x}}(f_\varepsilon)) = \tilde{\sigma}(f_{\overline{x}}) = f(\overline{x})$ for all $\overline{x} \in \Sigma^*$. ■

The above proposition establishes a crucial property that makes this theory appealing, as it means that the functions $f = f_\varepsilon$, $f_{\overline{x}} = \tilde{\tau}_{\overline{x}}(f)$, the linear operators $\tilde{\tau}_z$ and the linear functional $\tilde{\sigma}$ have coordinate representations as vectors and matrices with respect to some basis $\mathcal{B}$ for $\mathcal{F}$. Note that this remains true even if $\operatorname{rank}(f) = \infty$, but the coordinate representations will then be infinite and of little practical use. The property $f(\overline{x}) = \tilde{\sigma}(\tilde{\tau}_{\overline{x}}(f_\varepsilon))$ (cf. Equation 1) means that the function f is fully described by the data $(\tilde{\sigma}, \{\tilde{\tau}_z\}, f_\varepsilon)$. If these are given in some coordinate representation, then we have a SS representation:

Proposition 2 *Let $f : \Sigma^* \to K$ be given. If* $\operatorname{rank}(f) = d < \infty$*, then there exists a d-dimensional SS $\mathcal{M}$ such that $f = f_{\mathcal{M}}$.*

Proof Let $\mathcal{B}$ be a basis for $\mathcal{F}$, and let $\mathcal{M} = (\sigma, \{\tau_z\}, \omega_\varepsilon)$ be the coordinate representations of $(\tilde{\sigma}, \{\tilde{\tau}_z\}, f_\varepsilon)$ with respect to $\mathcal{B}$, where we are using the definitions for $\tilde{\sigma}$ and $\tilde{\tau}_z$ from the above Proposition 1. Then for any $\overline{x} \in \Sigma^*$, we have $f(\overline{x}) = \tilde{\sigma}(\tilde{\tau}_{\overline{x}}(f_\varepsilon)) = \sigma\tau_{\overline{x}}\omega_\varepsilon = f_{\mathcal{M}}(\overline{x})$. ■

Note that for the SS $\mathcal{M}$ constructed in Proposition 2 as a coordinate representation with respect to some basis $\mathcal{B}$ of $\mathcal{F}$, the states $\omega_{\overline{x}} = \tau_{\overline{x}}\omega_\varepsilon$ will be the coordinate representations of the functions $f_{\overline{x}} = \tilde{\tau}_{\overline{x}}(f)$ with respect to the basis $\mathcal{B}$. Also note that due to Equation (1) we may evaluate $f(\overline{x})$ using the SS $\mathcal{M}$ without knowledge of the basis $\mathcal{B}$.

The above proposition suggests that two SS might describe the same function f if and only if they are representations for f with respect to different bases for $\mathcal{F}$. However, this is only correct for so-called *minimal* SS, as will be detailed out in the following.

Definition 3 *Two SSs $\mathcal{M}$ and $\mathcal{M}'$ are* equivalent*, denoted by $\mathcal{M} \cong \mathcal{M}'$, if they define the same function, i.e., if $f_{\mathcal{M}} = f_{\mathcal{M}'}$. It is clear that this notion is an equivalence relation on the set of all SSs.*

We now introduce concepts needed to characterize the equivalence on SS. We give such a characterization for *minimal* SS in Proposition 12. For this, we introduce the concept of minimal SS, give a criteria for minimality in Corollary 8 and a procedure in Algorithm 2 to construct an equivalent minimal SS.

Definition 4 *For a given SS $\mathcal{M}$ we call the linear spaces $W = \operatorname{span}\{\tau_{\overline{x}}\omega_\varepsilon \,|\, \overline{x} \in \Sigma^*\}$ the* state space *and $\tilde{W} = \operatorname{span}\{(\sigma\tau_{\overline{x}})^\top \,|\, \overline{x} \in \Sigma^*\}$ the* co-state space *of $\mathcal{M}$.*

Definition 5 *We call a d-dimensional SS $\mathcal{M}$* trimmed *if it has full state and co-state spaces, i.e., if $W = \tilde{W} = K^d$. We call a SS* minimal *if no equivalent model of lower dimension exists.*

It will turn out in Corollary 8 that a SS is minimal if and only if it is trimmed. But first, we show how to construct bases for the state (and co-state) space of a given SS.

Proposition 6 *The following procedure constructs a basis $\mathcal{B}$ for the state space W of a given d-dimensional SS in time $\mathcal{O}(\max\{d, |\Sigma|\}d^3)$ (the construction of a basis $\tilde{\mathcal{B}}$ for the co-state space $\tilde{W}$ is analogous):*

Algorithm 1: Compute a basis $\mathcal{B}$ for the state space W of a given SS

```
B ← {}, C ← {ω_ε}
while |C| > 0 do
    ω ← some element of C, C ← C \ {ω}
    if ω is linearly independent of B then
        B ← B ∪ {ω}
        C ← C ∪ {τ_z ω | z ∈ Σ}
```

Proof At any time during the run of the algorithm, $\mathcal{B}$ is a set of linearly independent vectors. Furthermore the set $\mathcal{C}$ of "candidate vectors" increases by $|\Sigma|$ elements each time a new vector is added to the set $\mathcal{B}$, but decreases by one element each run through the main loop. Therefore, the algorithm terminates after at most $d|\Sigma| + 1$ runs through the main loop, since there are at most d linearly independent vectors that can be added to $\mathcal{B}$. Next we examine the runtime of the algorithm. Checking ω for linear independence from $\mathcal{B}$ can be done by checking $P_{\mathcal{B}}\omega = \omega$ in time $\mathcal{O}(d^2)$ if the orthogonal projection matrix $P_{\mathcal{B}}$ onto $\operatorname{span}(\mathcal{B})$ is known. This check is performed at most $d|\Sigma|+1$ times, yielding a complexity of $\mathcal{O}(d^3|\Sigma|)$. Clearly, the matrix $P_{\mathcal{B}}$ must be updated every time a vector is added to $\mathcal{B}$, which is a $\mathcal{O}(d^3)$ operation that needs to performed at most d times, giving a total complexity of $\mathcal{O}(d^4)$. Finally, every time a vector ω is added to $\mathcal{B}$, the set $\mathcal{C}$ is increased by $|\Sigma|$ vectors, each of which requires time $\mathcal{O}(d^2)$ to be computed from ω, for a total time complexity of $\mathcal{O}(d^3|\Sigma|)$. Adding these together gives the claimed time complexity.

Finally, we show that the returned set $\mathcal{B}$ is indeed a basis of the state-space W. Observe that for all $\omega \in \mathcal{B}$ and for all $z \in \Sigma$, the vectors $\tau_z\omega$ have been added as "candidate vectors" to the set $\mathcal{C}$ at some point during the run of the algorithm — namely when ω was added to $\mathcal{B}$. Each of these vectors is checked in turn and is at that point either linearly dependent on $\mathcal{B}$, or added to $\mathcal{B}$. Therefore, these vectors $\tau_z\omega$ are all linearly dependent on the final set $\mathcal{B}$, i.e., $\tau_z(\mathcal{B}) \subseteq \operatorname{span}(\mathcal{B})$ for all $z \in \Sigma$. By linearity of τ_z this implies that also

$\tau_z(\mathrm{span}(\mathcal{B})) \subseteq \mathrm{span}(\mathcal{B})$ for all $z \in \Sigma$. So $\mathrm{span}(\mathcal{B})$ contains ω_ε and is closed under the action of τ_z for all $z \in \Sigma$, which implies that $\{\tau_{\overline{x}}\omega_\varepsilon \,|\, \overline{x} \in \Sigma^*\} \subseteq \mathrm{span}(\mathcal{B})$. But $\mathcal{B} \subset \{\tau_{\overline{x}}\omega_\varepsilon \,|\, \overline{x} \in \Sigma^*\}$ by construction of $\mathcal{B}$. Together, this implies $\mathrm{span}(\mathcal{B}) = \mathrm{span}(\{\tau_{\overline{x}}\omega_\varepsilon \,|\, \overline{x} \in \Sigma^*\}) = W$. ■

The above is a polynomial time algorithm for which we have explicitly stated the runtime complexity, since it is the workhorse for the operations of this section and dominates their runtimes. Note further that the computed bases are by construction of the form $\mathcal{B} = \{\tau_{\overline{x}_j}\omega_\varepsilon \,|\, j \in J\}$ and $\tilde{\mathcal{B}} = \{(\sigma\tau_{\overline{x}_i})^\top \,|\, i \in I\}$ for suitable index sets I, J and corresponding words $\overline{x}_i$ and $\overline{x}_j$ of length at most d, where d is the dimension of the SS. Also, the above procedure allows us to check whether a given SS is trimmed.

The following proposition is the core technical result needed to establish the connection between a SS being trimmed, having full rank, and being minimal.

Proposition 7 *For a d-dimensional SS $\mathcal{M}$, let $\{\tau_{\overline{x}_j}\omega_\varepsilon \,|\, j \in J\}$ and $\{(\sigma\tau_{\overline{x}_i})^\top \,|\, i \in I\}$ be bases for W and $\tilde{W}$ respectively, and define $F^{I,J} = [f_{\mathcal{M}}(\overline{x}_j\overline{x}_i)]_{(i,j)\in I\times J}$, then* $\mathrm{rank}(\mathcal{M}) = \mathrm{rank}(F^{I,J}) \leq \min\{|I|, |J|\} \leq d$. *Furthermore, if $|I| = d$ or $|J| = d$ then* $\mathrm{rank}(\mathcal{M}) = \min\{|I|, |J|\}$.

Proof Define $\Pi = ((\sigma\tau_{\overline{x}_k})^\top)_{k\in\mathbb{N}}^\top$ and $\Phi = (\tau_{\overline{x}_k}\omega_\varepsilon)_{k\in\mathbb{N}}$, as well as $\Pi_I = ((\sigma\tau_{\overline{x}_i})^\top)_{i\in I}^\top \in K^{|I|\times d}$ and $\Phi_J = (\tau_{\overline{x}_j}\omega_\varepsilon)_{j\in J} \in K^{d\times|J|}$. The rows of Π_I are a basis for the row space of Π and the columns of Φ_J are a basis for the column space of Φ. Now $F = \Pi\Phi$ and $F^{I,J} = \Pi_I\Phi_J$. Therefore $\mathrm{rank}(\mathcal{M}) := \mathrm{rank}(F) = \mathrm{rank}(\Pi\Phi) = \mathrm{rank}(\Pi_I\Phi) = \mathrm{rank}(\Pi_I\Phi_J) = \mathrm{rank}(F^{I,J})$. Moreover, $\mathrm{rank}(\Pi_I) = |I|$ and $\mathrm{rank}(\Phi_J) = |J|$ imply that $\mathrm{rank}(\Pi_I\Phi_J) \leq \min\{|I|, |J|\} \leq d$ as well as $\mathrm{rank}(\Pi_I\Phi_J) = |J|$ if $|I| = d$ and $\mathrm{rank}(\Pi_I\Phi_J) = |I|$ if $|J| = d$. ■

From this, we obtain the following result, which allows us to check a d-dimensional SS for minimality by checking whether the SS is trimmed, i.e., by constructing bases for the state and co-state space and checking if these have dimension d.

Corollary 8 *Let $\mathcal{M}$ be a d-dimensional SS. The following are equivalent:*

(i) *$\mathcal{M}$ is trimmed*

(ii) $\mathrm{rank}(\mathcal{M}) = d$

(iii) *$\mathcal{M}$ is minimal*

Proof If $\mathcal{M}$ has full rank, i.e., $\mathrm{rank}(\mathcal{M}) = d$, then $\mathcal{M}$ must be minimal, as any lower-dimensional SS must have a lower rank and therefore cannot be equivalent. Conversely, if $\mathcal{M}$ is minimal, then we must have $\mathrm{rank}(\mathcal{M}) = d$, since by Proposition 2 there exists a $\mathrm{rank}(\mathcal{M})$-dimensional equivalent SS. By Proposition 7 — and using the notation from the proposition — we see that $\mathrm{rank}(\mathcal{M}) = d \Leftrightarrow |I| = |J| = d$, i.e., if and only if $\mathcal{M}$ is trimmed. ■

Next, we define the transformation of a SS by linear maps ρ and ρ'. Such transformations will serve as the basic operation on SS for all conversion operations.

Definition 9 *For a d-dimensional SS $\mathcal{M} = (\sigma, \{\tau_z\}, \omega_\varepsilon)$ and any matrices $\rho \in K^{n\times d}$ and $\rho' \in K^{d\times n}$, we define the n-dimensional SS $\rho\mathcal{M}\rho' := (\sigma\rho', \{\rho\tau_z\rho'\}, \rho\omega_\varepsilon)$.*

If ρ is non-singular, and $\rho' = \rho^{-1}$, then this transformation will yield an equivalent *conjugated* SS. If the SS is minimal, then this corresponds to a change of basis for the underlying function space $\mathcal{F}$.

Lemma 10 *Let $\mathcal{M} = (\sigma, \{\tau_z\}, \omega_\varepsilon)$ be a d-dimensional SS, and $\rho \in \mathbb{R}^{d\times d}$ be non-singular. Then $\mathcal{M} \cong \rho\mathcal{M}\rho^{-1}$. We will call $\rho\mathcal{M}\rho^{-1}$ a* conjugate *of $\mathcal{M}$.*

Proof $\forall \overline{x} \in \Sigma^* : f_{\rho\mathcal{M}\rho^{-1}}(\overline{x}) = (\sigma\rho^{-1})(\rho\tau_{x_n}\rho^{-1})\cdots(\rho\tau_{x_1}\rho^{-1})(\rho\omega_\varepsilon) = \sigma\tau_{\overline{x}}\omega_\varepsilon = f_{\mathcal{M}}(\overline{x})$. ■

We already know how to check for minimality. We now show how to convert a given SS to an equivalent minimal SS using the introduced transformations on SSs.

Proposition 11 *For a given SS $\mathcal{M}$, the following procedure constructs an equivalent minimal SS $\mathcal{M}''$:*

Algorithm 2: Minimization of a SS $\mathcal{M}$

1 *Construct a basis $\{\tau_{\overline{x}_j}\omega_\varepsilon \mid j \in J\}$ for the state space W of $\mathcal{M}$*
Set $\Phi = (\tau_{\overline{x}_j}\omega_\varepsilon)_{j\in J}$.
Set $\mathcal{M}' = \Phi^\dagger\mathcal{M}\Phi$, where $\Phi^\dagger$ denotes the Moore-Penrose pseudoinverse of Φ.
2 *Construct a basis $\{(\sigma'\tau'_{\overline{x}_i})^\top \mid i \in I'\}$ for the co-state space $\tilde{W}'$ of $\mathcal{M}'$.*
Set $\Pi' = ((\sigma'\tau'_{\overline{x}_i})^\top)^\top_{i\in I'}$.
Set $\mathcal{M}'' = \Pi'\mathcal{M}'\Pi'^\dagger$.

Proof Note that by construction the columns of Φ and $\Pi'^\top$ form bases for the spaces W and $\tilde{W}'$ respectively. Therefore, $\Phi^\dagger\Phi = id$ and $\Phi\Phi^\dagger|_W = id$, as well as $(\Pi'^\top)^\dagger\Pi'^\top = id$ and $\Pi'^\top(\Pi'^\top)^\dagger|_{\tilde{W}'} = id$. We can simply check equivalence, i.e., that for any $\overline{x} \in \Sigma^*$,

$$\begin{aligned}
f_{\mathcal{M}''}(\overline{x}) &= \sigma''\tau''_{x_n}\cdots\tau''_{x_1}\omega''_\varepsilon \\
&= \sigma'\Pi'^\dagger\Pi'\tau'_{x_n}\Pi'^\dagger\cdots\Pi'\tau'_{x_1}\Pi'^\dagger\Pi'\omega'_\varepsilon \\
&= \omega'^\top_\varepsilon\Pi'^\top(\Pi'^\top)^\dagger\tau'^\top_{x_1}\Pi'^\top\cdots(\Pi'^\top)^\dagger\tau'^\top_{x_n}\Pi'^\top(\Pi'^\top)^\dagger\sigma'^\top \\
&= \sigma'\tau'_{x_n}\cdots\tau'_{x_1}\omega'_\varepsilon \\
&= \sigma\Phi\Phi^\dagger\tau_{x_n}\Phi\cdots\Phi^\dagger\tau_{x_1}\Phi\Phi^\dagger\omega_\varepsilon \\
&= \sigma\tau_{\overline{x}}\omega_\varepsilon = f_{\mathcal{M}}(\overline{x}).
\end{aligned}$$

Next, consider $(\tau'_{\overline{x}_j}\omega'_\varepsilon)_{j\in J} = (\Phi^\dagger\tau_{\overline{x}_j}\omega_\varepsilon)_{j\in J} = \Phi^\dagger\Phi = id$. This implies that $\mathcal{M}'$ has full state space W' and that $\{\tau'_{\overline{x}_j}\omega'_\varepsilon \mid j \in J\}$ is a basis for W', since the dimension d' of $\mathcal{M}'$ is $|J|$ by construction. By Proposition 7, $|J| = d'$ implies $\mathrm{rank}(\mathcal{M}') = \min(|I'|, |J|) = |I'|$. By construction $|I'| = d''$ where d'' is the dimension of $\mathcal{M}''$. Furthermore, $\mathrm{rank}(\mathcal{M}') = \mathrm{rank}(\mathcal{M}'')$ since $\mathcal{M}' \cong \mathcal{M}''$ so by Corollary 8 $\mathcal{M}''$ is minimal. ■

As we can convert any SS to an equivalent minimal SS using the above Algorithm 2, it will be sufficient to characterize equivalence only for minimal SS. This is done by the following result.

Proposition 12 *Let $\mathcal{M} = (\sigma, \{\tau_z\}, \omega_\varepsilon)$ and $\mathcal{M}' = (\sigma', \{\tau'_z\}, \omega'_\varepsilon)$ be minimal d-dimensional SS. The following are equivalent:*

(i) $\mathcal{M} \cong \mathcal{M}'$

(ii) $\mathcal{M}' = \rho\mathcal{M}\rho^{-1}$ *for some non-singular* $\rho \in K^{d\times d}$

(iii) $\Pi\Phi = \Pi'\Phi'$, $\Pi\omega_\varepsilon = \Pi'\omega'_\varepsilon$, $\sigma\Phi = \sigma'\Phi'$ *and* $\forall z \in \Sigma : \Pi\tau_z\Phi = \Pi'\tau'_z\Phi'$, *where* $\{\tau_{\overline{x}_j}\omega_\varepsilon \mid j \in J\}$ *and* $\{(\sigma\tau_{\overline{x}_i})^\top \mid i \in I\}$ *are bases for the state and co-state spaces W and $\tilde{W}$ of $\mathcal{M}$ respectively, and* $\Pi = ((\sigma\tau_{\overline{x}_i})^\top)_{i\in I}^\top$, $\Phi = (\tau_{\overline{x}_j}\omega_\varepsilon)_{j\in J}$, $\Pi' = ((\sigma'\tau'_{\overline{x}_i})^\top)_{i\in I}^\top$, *and* $\Phi' = (\tau'_{\overline{x}_j}\omega'_\varepsilon)_{j\in J}$.

Proof Lemma 10 establishes $(ii) \Rightarrow (i)$. For $(i) \Rightarrow (iii)$ note that $f_\mathcal{M} = f_{\mathcal{M}'}$ implies that $\Pi\tau_{\bar{z}}\Phi = [f(\overline{x}_j\bar{z}\overline{x}_i)]_{i,j\in I\times J} = \Pi'\tau'_{\bar{z}}\Phi'$ for all $\bar{z} \in \Sigma^*$, as well as $\Pi\omega_\varepsilon = (f(\overline{x}_i))_{i\in I}^\top = \Pi'\omega'_\varepsilon$ and $\sigma\Phi = (f(\overline{x}_j))_{j\in J} = \sigma'\Phi'$. Finally, to see $(iii) \Rightarrow (ii)$, note that Π and Φ have full rank, since $\mathcal{M}$ is minimal, so Π' and Φ' must also have full rank. Let $\rho = \Pi'^{-1}\Pi = \Phi'\Phi^{-1}$, then $\rho^{-1} = \Phi\Phi'^{-1}$. We can now easily check that $\mathcal{M}' = \rho\mathcal{M}\rho^{-1}$. ■

Note that this allows us to decide equivalence for any given SS $\mathcal{M}$ and $\mathcal{M}'$ by first converting them to equivalent minimal SS $\tilde{\mathcal{M}}$ and $\tilde{\mathcal{M}}'$ respectively using Algorithm 2, and then checking for equivalence by criteria (iii) from the above Proposition 12. The required bases for the state and co-state spaces of $\tilde{\mathcal{M}}$ and $\tilde{\mathcal{M}}'$ can be computed by Algorithm 1.

The following proposition shows that any SS can be transformed into an equivalent SS where σ and ω_ε can be essentially any desired vectors. This implies that it is no restriction to assume some fixed form for σ, as is sometimes done. For instance, in the case of OOMs often $\sigma = (1, \ldots, 1)$ is used, while for MA often $\sigma = (1, 0, \ldots, 0)$ is assumed.

Proposition 13 *Let $\mathcal{M} = (\sigma, \{\tau_z\}, \omega_\varepsilon)$ be a d-dimensional SS, and let $\sigma'^\top, \omega'_\varepsilon \in K^d$ such that $\sigma'\omega'_\varepsilon = \sigma\omega_\varepsilon$. Then there exists a non-singular linear map ρ such that $\rho\mathcal{M}\rho^{-1} = (\sigma', \{\tau'_z\}, \omega'_\varepsilon)$.*

Proof Extend $\{\sigma^\top\}$ to an orthogonal basis $\{\sigma^\top, v_2, \ldots, v_d\}$ of K^d, and $\{\sigma'^\top\}$ to an orthogonal basis $\{\sigma'^\top, v'_2, \ldots, v'_d\}$ of K^d. We distinguish two cases:

If $c := \sigma\omega_\varepsilon = \sigma'\omega'_\varepsilon \neq 0$, then $\rho_1 = (\omega_\varepsilon, v_2, \ldots, v_d)^{-1}$ and $\rho_2 = (\omega'_\varepsilon, v'_2, \ldots, v'_d)$ are non-singular. Let $\rho = \rho_2\rho_1$. We can easily see that $\rho_2\rho_1\omega_\varepsilon = \rho_2 e_1 = \omega'_\varepsilon$ and $\sigma\rho^{-1} = \sigma\rho_1^{-1}\rho_2^{-1} = c \cdot e_1^\top\rho_2^{-1} = \sigma'$, since $\sigma'\rho_2 = c \cdot e_1^\top$.

If $\sigma\omega_\varepsilon = \sigma'\omega'_\varepsilon = 0$, then (perhaps after reordering v_i and v'_i) $\rho_1 = (\frac{\sigma^\top}{\sigma\sigma^\top}, \omega_\varepsilon, v_3, \ldots, v_d)^{-1}$ and $\rho_2 = (\frac{\sigma'^\top}{\sigma'\sigma'^\top}, \omega'_\varepsilon, v'_3, \ldots, v'_d)$ are non-singular. Let $\rho = \rho_2\rho_1$. We can again check that $\rho_2\rho_1\omega_\varepsilon = \rho_2 e_2 = \omega'_\varepsilon$ and $\sigma\rho^{-1} = \sigma\rho_1^{-1}\rho_2^{-1} = e_1\rho_2^{-1} = \sigma'$, since $\sigma'\rho_2 = e_1$. ■

Finally, we introduce a special property called *interpretability* that a SS can have. This concept has led to some confusion in the past — especially regarding the relationship between OOMs and PSRs. This is due to the fact that it has been defined differently for OOMs, IO-OOMs and PSRs, as will be discussed later. Another source of confusion is that interpretability has been regarded as a crucial property for learning, which is however only

true for the the very early learning algorithms. Here we give a definition of interpretability that works for all models, and we will defer the discussion of the different uses to the later sections.

Definition 14 *A d-dimensional SS $\mathcal{M}$ is said to be* interpretable w.r.t. the sets $Y_1, \ldots, Y_d \subset \Sigma^*$ *if the states $\omega_{\overline{x}}$ take the form $\omega_{\overline{x}} = [f_{\mathcal{M}}(\overline{x}Y_1), \ldots, f_{\mathcal{M}}(\overline{x}Y_d)]^\top$ for all $\overline{x} \in \Sigma^*$, where $f_{\mathcal{M}}(\overline{x}Y) = \sum_{\overline{y}\in Y} f_{\mathcal{M}}(\overline{xy})$.*

The following proposition and algorithm show how to *make a SS interpretable*, i.e., how to convert any given SS into an equivalent interpretable form.

Proposition 15 *Let $\mathcal{M} = (\sigma, \{\tau_z\}, \omega_\varepsilon)$ be a d-dimensional minimal SS, and $Y_1, \ldots, Y_d \subset \Sigma^*$. If $\rho = [(\sigma\tau_{Y_1})^\top, \ldots, (\sigma\tau_{Y_d})^\top]^\top$ is non-singular, where $\tau_Y = \sum_{\overline{y}\in Y} \tau_{\overline{y}}$, then $\mathcal{M}' := \rho\mathcal{M}\rho^{-1} \cong \mathcal{M}$ and $\mathcal{M}'$ is interpretable w.r.t. $Y_1, \ldots, Y_d$.*

Proof $\forall \overline{x} \in \Sigma^* : \omega'_{\overline{x}} = \rho\omega_{\overline{x}} = [\sigma\tau_{Y_1}\tau_{\overline{x}}\omega_\varepsilon, \ldots, \sigma\tau_{Y_d}\tau_{\overline{x}}\omega_\varepsilon]^\top = [f_{\mathcal{M}}(\overline{x}Y_1), \ldots, f_{\mathcal{M}}(\overline{x}Y_d)]^\top$. ■

Corollary 16 *For a SS $\mathcal{M}$, the following algorithm returns an equivalent interpretable SS.*

Algorithm 3: Make a SS $\mathcal{M}$ of rank d interpretable

1 *Minimize $\mathcal{M}$, i.e., find an equivalent minimal SS $\mathcal{M}'$ using Algorithm 2.*
2 *Construct a basis $\{(\sigma'\tau'_{\overline{x}_i})^\top \mid i \in I\}$ of the co-state space $\tilde{W}'$ of $\mathcal{M}'$ using Algorithm 1*
Select sets $Y_k = \{\overline{x}_{i_k}\}$ where $\{i_1, \ldots, i_d\} = I$.
Set $\rho = [(\sigma'\tau'_{Y_1})^\top, \ldots, (\sigma'\tau'_{Y_d})^\top]^\top$.
3 *Return $\rho\mathcal{M}'\rho^{-1}$.*

Proof The above algorithm indeed returns an equivalent SS that is interpretable w.r.t. the selected sets Y_k, since $\mathcal{M}'$ is minimal and therefore ρ is non-singular by construction. ■

3. Versions of Sequential Systems

In this section we first show that SS are an algebraic characterization of multiplicity automata (MA), and we mention the relationship to the more general class of weighted finite automata (WFA) and its extension to weighted finite-state transducers (WFST). We then define stochastic multiplicity automata (SMA), observable operator models (OOMs) and predictive state representations (PSRs), which are known to generalize probabilistic finite automata (PFA), hidden Markov models (HMMs) and partially observable Markov decision processes (POMDPs), respectively. We show that these are all instances of SSs that are used to model different kinds of objects. Furthermore, we examine the relations between these models. An overview is given in Figure 1.

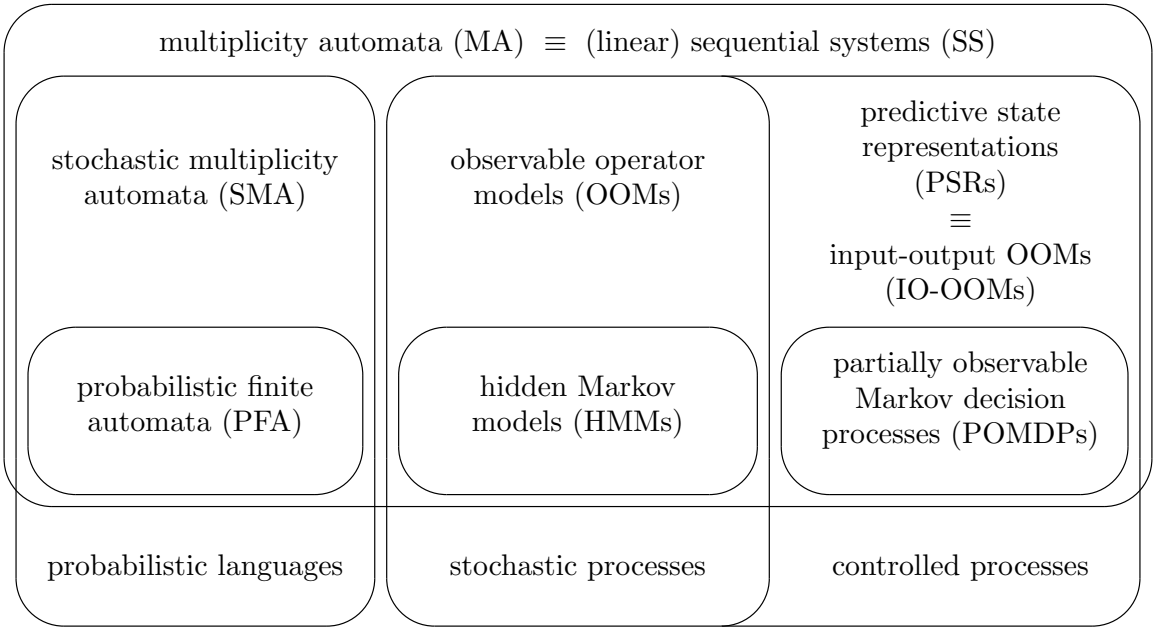


Figure 1: SMA, OOMs and PSRs are versions of SSs that model probabilistic languages, stochastic processes and controlled processes respectively, and strictly generalize PFA, HMMs and POMDPs respectively.

3.1 Multiplicity Automata and Weighted Automata

The above definition of linear finite dimensional SS is an equivalent algebraic way of looking at a type of automata that were introduced by Schützenberger (1961) and are most commonly known as *multiplicity automata* (Salomaa and Soittola, 1978; Berstel and Reutenauer, 1988). We will give a very brief introduction.

Definition 17 *A* K-multiplicity automaton *(MA) is a structure* $\langle \Sigma, Q, \varphi, \iota, \tau \rangle$*, where* Σ *is an* alphabet*,* Q *is a finite set of* states*,* $\varphi : Q \times \Sigma \times Q \to K$ *is the* state transition function*,* $\iota : Q \to K$ *is the* initialization function*, and* $\tau : Q \to K$ *is the* termination function*. The state transition function is extended to words by setting* $\forall \overline{x} \in \Sigma^*, z \in \Sigma$: $\varphi(q, \overline{x}z, q') = \sum_{s \in Q} \varphi(q, \overline{x}, s)\varphi(s, z, q')$*, and* $\varphi(q, \varepsilon, q') = 1$ *if* $q = q'$ *and 0 otherwise. A multiplicity automaton* $\mathcal{M}$ *then defines a function*

$$f_{\mathcal{M}} : \Sigma^* \to K, \quad f_{\mathcal{M}}(\overline{x}) = \sum_{q,q' \in Q} \iota(q)\varphi(q, \overline{x}, q')\tau(q').$$

The formal equivalence of MA to linear finite-dimensional SS is easily seen by rewriting the definition of MA in terms of matrix multiplication: Set $\omega_\varepsilon = [\iota(q_i)]_i$, $\tau_z = [\varphi(q_j, z, q_i)]_{i,j}$, and $\sigma = [\tau(q_j)]_j^\top$. Then we have $\tau_{\overline{x}z} = [\varphi(q_j, \overline{x}z, q_i)]_{i,j} = [\sum_{q_k \in Q} \varphi(q_j, \overline{x}, q_k)\varphi(q_k, z, q_i)]_{i,j} = [\varphi(q_k, z, q_i)]_{i,k}[\varphi(q_j, \overline{x}, q_k)]_{k,j} = \tau_z\tau_{\overline{x}}$ and similarly $f_{\mathcal{M}}(\overline{x}) = \sigma\tau_{\overline{x}}\omega_\varepsilon$. However, the above definition of MA makes it apparent how MA are an extension of non-deterministic finite

automata (NFA) to WFA that add weights to the initial and terminal states as well as the state transitions. The weight of a path from an initial state to a termination state is then given by the product of the corresponding weights (hence the name *multiplicity* automata), while the value $f_{\mathcal{M}}(\overline{x})$ is computed by summing the weights of all paths compatible with $\overline{x}$.

At this point we should mention that MA as defined here are merely a special case of WFA. The difference is that for MA we consider weights from a field K (here $K = \mathbb{R}$ or $K = \mathbb{C}$), while for WFA the weights are only required to come from an algebraic structure K called a semiring. There exists a large body of theory for WFA that generalizes the theory of SS that we have presented in Section 2, which can be found in the recent textbook by Droste et al. (2009). Note that while MA and WFA are formally closely related, there is a difference in the way they are viewed and used. For instance, WFA are often considered over the semiring $\mathbb{R}^+$ with weights given the interpretation of transition probabilities, which are then called probabilistic finite automata (PFA). Such PFA are graphical models, and the states Q are latent states. For $\mathbb{R}$-MA, however, the weights are allowed to be negative, and the weights as well as the states Q become abstract notions. In other words, PFA (and WFA in general) are typically used when the states and transition structure carry some meaning, while MA are typically used as an abstract tool to characterize functions $f : \Sigma^* \to K$. This difference in perspective is reflected in the relationship of PFA to SMA, HMM to OOM and POMDP to PSR described in the remainder of this Section 3. Note that PFA are a special case of MA, as $\mathbb{R}^+ \subset \mathbb{R}$. In fact, there exist functions $f : \Sigma^* \to \mathbb{R}^+$ that can be described by a MA, but not by a PFA, i.e., MA are strictly more general than PFA. This sequence of increasing generalization starting with finite automata (FA) can be summarized as follows:

$$\text{FA} \subset \text{NFA} \subset \text{PFA} \subset \text{MA} \equiv \text{SS} \subset \text{WFA}.$$

Furthermore, there exists a natural extension of WFA to input-output systems that are called weighted finite-state transducers (WFST). Here, the alphabet Σ is split as $\Sigma = \Sigma_I \times \Sigma_O$, where Σ_I is regarded as input alphabet and Σ_O as output alphabet. The function $f_{\mathcal{M}} : \Sigma_I^* \times \Sigma_O^* \to K$ is then viewed as describing a relation between Σ_I and Σ_O. Again, K is in general only required to be a semiring, but a typical choice is $K = \mathbb{R}^+$ with the interpretation of state transition probabilities, yielding a latent variable model called probabilistic finite-state transducers (PFST). WFST are a flexible class of models that have been shown to unify several common approaches used in the the context of language and speech processing; a survey is given by Mohri et al. (2002). Furthermore, IO-OOMs and thereby PSRs (cf. Section 3.2 and Section 3.3) are in fact WFST with weights in $K = \mathbb{R}$, although they are not usually viewed this way, as WFST are typically seen as latent variable models, while IO-OOMs and PSRs are not. However, since PFST are MA, as long as the desired application merely requires the characterization of the function $f_{\mathcal{M}} : \Sigma_I^* \times \Sigma_O^* \to \mathbb{R}^+$, the SS learning algorithms described in Section 4 can be applied to the case of WFST as well, as has been done recently by Balle et al. (2011).

Note that in the context of MA one is usually interested in characterizing functions $f : \Sigma^* \to K$, which are also called *formal series* in general and *recognizable series* if they are computed by a MA. However, a MA $\mathcal{M}$ can also be used to recognize a language $L \subseteq \Sigma^*$ by setting $L_{\mathcal{M}} = \{\overline{x} \in \Sigma^* \mid f_{\mathcal{M}}(\overline{x}) \subseteq J\}$ for some subset $J \subseteq K$, e.g., $J = \{k \in K : k > \kappa\}$

for some threshold parameter $\kappa \geq 0$. The class of languages recognizable by MA is known to be strictly more general than the class of regular languages (Cortes and Mohri, 2000).

MA have received a lot of attention in the context of learning theory following the discovery of efficient learning algorithms (Bergadano and Varricchio, 1994; Ohnishi et al., 1994) in an extended version of the exact learning model of Angluin (1987). This led to further results on the learnability of several classes of DNF formulae (Bergadano et al., 1996), the class of polynomials over finite fields, decision trees and others (Beimel et al., 1996, 2000).

3.1.1 Stochastic Multiplicity Automata and Stochastic Languages

Additionally, MA have been applied in the context of probabilistic grammatical inference (Denis et al., 2006; Bailly et al., 2009), which is of particular interest to us because of the close relationship of these approaches to OOMs and PSRs — as we shall see.

Definition 18 *A function $f : \Sigma^* \to \mathbb{R}$ that satisfies $0 \leq f \leq 1$ and $f(\Sigma^*) = \sum_{\overline{x} \in \Sigma^*} f(\overline{x}) = 1$ is called a* stochastic language*,* probabilistic language *or just* distribution *over Σ^*. A distribution $f_{\mathcal{M}}$ on Σ^* that is defined by some MA $\mathcal{M}$ is called a* rational stochastic language*, and a MA that defines such a distribution is called a* stochastic MA *(SMA).*

Denis and Esposito (2008) give a comprehensive overview of rational stochastic languages over various fields K, their relationships and relations to subclasses such as the important class of probabilistic regular languages.

Definition 19 *A* probabilistic (finite) automaton *(PFA) is a SMA with the following restrictions: (i) ι, τ, φ have values in $[0, 1]$, and (ii) $\iota(Q) = 1$ and $\forall q \in Q : \tau(q) + \varphi(q, \Sigma, Q) = 1$, where $\iota(Q) = \sum_{q \in Q} \iota(q)$ and $\varphi(q, \Sigma, Q) = \sum_{x \in \Sigma} \sum_{q' \in Q} \varphi(q, x, q')$. The stochastic languages that can be represented by PFA are called* probabilistic regular languages*.*

PFA are closely related to hidden Markov models (HMMs), and the relationship has been detailed out by Dupont et al. (2005). It is however less well known that SMA are closely related to observable operator models — a class of models for stochastic processes that generalize HMM in a similar way that SMA generalize PFA.

We point out two results that are relevant in the context of modeling probabilistic languages by MA. First of all, it is known that it is an NP-hard problem to compute the maximum likelihood estimate of parameters of a PFA with known structure from a given training set of words (Abe and Warmuth, 1992). In practice, algorithms based on expectation maximization (EM) (Dempster et al., 1977) are used which compute locally optimal models instead. In contrast to this, the algebraic theory for SSs allows for powerful learning algorithms (see Section 4) that often outperform EM-trained PFA or HMMs (Rosencrantz et al., 2004; Jaeger et al., 2006a). However, these learning algorithms may return MA that are arbitrarily close to SMA but fail to represent stochastic languages. It is in fact undecidable whether a MA represents a stochastic language (Denis and Esposito, 2004).

3.2 Observable Operator Models and Stochastic Processes

Observable operator models were introduced by Jaeger (1997) as a concise algebraic characterization of stochastic processes (see also Jaeger, 1998, 2000b; Jaeger et al., 2006b). These

models are closely related to other algebraic characterizations of stochastic processes (Heller, 1965; Ito, 1992; Upper, 1997) that were studied in the context of deciding the equivalence for HMMs (Gilbert, 1959), which came to a successful conclusion by framing HMMs in the more general class of *linearly dependent processes* by Ito et al. (1992).

Definition 20 *A* (discrete-valued) stochastic process *is a function* $f : \Sigma^* \to [0,1]$ *that satisfies (i)* $f(\varepsilon) = 1$ *and (ii)* $\forall \overline{x} \in \Sigma^* : f(\overline{x}) = \sum_{x \in \Sigma} f(\overline{x}x)$*. Such a function* f *defines the probabilities of initial observation sequences. An* observable operator model *(OOM) is a linear SS* $\mathcal{M}$ *such that* $f_{\mathcal{M}}$ *is a stochastic process. A stochastic process that can be modeled by a finite dimensional OOM is called a* linearly dependent process*.*

One of the interesting features of OOMs is their notion of "state" of a (stochastic) process. The idea that goes back to Zadeh (1969) is that a system state is really nothing more than the information that is required to predict the future. In the case of OOMs, the states $\omega_{\overline{x}}$ not only carry enough information to predict the future, they *are* (in a certain sense) just future predictions.

To see this, recall that the states $\omega_{\overline{x}}$ of a SS are coordinate representations of the functions $f_{\overline{x}}$ w.r.t. some unknown basis $\mathcal{B}$ of the function space $\mathcal{F}$. In the case of OOMs, these functions take on the meaning that $f_{\overline{x}}(\overline{y}) = P(\overline{x}\overline{y})$, i.e., they give the probability of observing the sequence $\overline{x}$ followed by $\overline{y}$. These functions are therefore called *future prediction functions* in the context of OOMs. The operators $\{\tau_z\}$ are then state update operators that update a state $\omega_{\overline{x}}$ (corresponding to the future prediction function $f_{\overline{x}}$ after an initial observation of $\overline{x}$) according to the new observation z to the new state $\omega_{\overline{x}z}$ (corresponding to the future prediction function $f_{\overline{x}z}$ after an initial observation of $\overline{x}z$) — hence the name "observable operators" (Jaeger, 1998).

For convenience, these functions $f_{\overline{x}}$, as well as the corresponding states $\omega_{\overline{x}}$, are often normalized to $f_{\overline{x}}/f(\overline{x})$ and $\omega_{\overline{x}}/\sigma\omega_{\overline{x}}$ respectively, since $f_{\overline{x}}(\overline{y})/f(\overline{x}) = \sigma\tau_{\overline{y}}\omega_{\overline{x}}/\sigma\omega_{\overline{x}} = P(\overline{y}|\overline{x})$, the probability of observing $\overline{y}$ given that $\overline{x}$ has been observed. Therefore, an OOM started in the normalized state $\omega_{\overline{x}}/\sigma\omega_{\overline{x}}$ represents a stochastic process started after an initial observation of $\overline{x}$. This corresponds to the notion of a *residual automaton* in the context of SMA, which is obtained by starting a SMA in the (normalized) state $\omega_{\overline{x}}/\sum_{\overline{z} \in \Sigma^*} \sigma\tau_{\overline{z}}\omega_{\overline{x}}$ and then represents a *residual language* (Denis and Esposito, 2004).

3.2.1 Relation to Hidden Markov Models

Any HMM can be trivially converted into an OOM. A hidden Markov model (HMM) consists of an unobserved Markov process X_t that takes values in a finite set of states $Q = \{s_1, \ldots, s_n\}$, and is governed by a stochastic state transition matrix $T = [P(X_{t+1} = s_j \,|\, X_t = s_i)]_{i,j}$. At each time step an observation Y_t from Σ is made according to the emission vector $E_z = [P(Y_t = z \,|\, X_t = s_i)]_i$. Finally, an initial state vector $\pi = [P(X_0 = s_i)]_i$ is needed to fully specify the distribution of the stochastic process Y_t (Rabiner, 1989).

Proposition 21 *(Jaeger, 2000b) A given HMM* $(T, \{E_z\}_{z \in \Sigma}, \pi)$ *with* N *states is equivalent to the OOM* $(\sigma, \{\tau_z\}, \omega_\varepsilon)$ *defined by* $\sigma = (1, \ldots, 1)$*,* $\tau_z = T^\top \mathrm{diag}(E_z)$ *and* $\omega_\varepsilon = \pi$*. The rank of the OOM is less than or equal to* N*.*

Moreover, there are examples of OOMs of finite rank that cannot be modeled by any HMM with a finite number of states. A prototypical example is the so-called "probability clock" (Jaeger, 1998). It is an open question how to find a "close" HMM for a given OOM. While OOMs can be seen as a generalization of HMMs, one should keep in mind that there is a fundamental difference in the notion of the state of the process. The state vector in the case of a HMM is a stochastic vector that expresses the belief about the underlying hidden state, while for an OOM it is a coordinate representation of the corresponding future prediction function. However, under certain conditions it is possible to recover HMM-like hidden states from an OOM (Hsu et al., 2009; Anandkumar et al., 2012).

3.2.2 Relationship to Stochastic Multiplicity Automata

The main difference between OOMs and SMA is that OOMs model stochastic processes, while SMA model distributions on words. However, we can use a stochastic process to model a distribution on words if we introduce a termination symbol \$.

Definition 22 *An OOM $\mathcal{M}$ over the alphabet $\Sigma_\$ = \Sigma \cup \{\$\}$ is* terminating *if $f_\mathcal{M}(\Sigma^*\$) := \sum_{\overline{x}\in\Sigma^*} \sigma\tau_\$\tau_{\overline{x}}\omega_\varepsilon = 1$.*

Proposition 23 *An OOM $\mathcal{M} = (\sigma, \{\tau_z\}, \omega_\varepsilon)$ over the alphabet Σ can be extended to a terminating OOM $\mathcal{M}' = (\sigma, \{\tau'_z\}, \omega_\varepsilon)$ over the alphabet $\Sigma_\$ = \Sigma\cup\{\$\}$ by setting $\tau'_z = (1-p)\tau_z$ and $\tau'_\$ = p\tau_\Sigma$ for some fixed termination probability $p \in (0,1)$, where $\tau_\Sigma = \sum_{z\in\Sigma}\tau_z$.*

Proof We first show that $\mathcal{M}'$ describes a stochastic process. Clearly, $f_{\mathcal{M}'} \geq 0$ and $f_{\mathcal{M}'}(\varepsilon) = \sigma\omega_\varepsilon = 1$. To show property (ii), take any $\overline{x} \in \Sigma^*_\$$ and note that by linearity $\tau'_{\overline{x}}\omega_\varepsilon = \sum_k \lambda_k \tau_{\overline{x}_k}\omega_\varepsilon$ for suitable $\lambda_k \in \mathbb{R}$ and sequences $\overline{x}_k \in \Sigma^*$ (this is obtained by replacing all occurrences of $\tau'_\$$ by $p\sum_{z\in\Sigma}\tau_z$). Then $\sum_{z\in\Sigma_\$} f_{\mathcal{M}'}(\overline{x}z) = \sigma(\sum_{z\in\Sigma_\$}\tau'_z)\tau'_{\overline{x}}\omega_\varepsilon = \sigma\tau_\Sigma\tau'_{\overline{x}}\omega_\varepsilon = \sum_k \lambda_k\sigma\tau_\Sigma\tau_{\overline{x}_k}\omega_\varepsilon = \sum_k \lambda_k\sigma\tau_{\overline{x}_k}\omega_\varepsilon = \sigma\tau'_{\overline{x}}\omega_\varepsilon = f_{\mathcal{M}'}(\overline{x})$. Furthermore, $f_{\mathcal{M}'}(\Sigma^*\$) = \sum_{\overline{x}\in\Sigma^*}\sigma\tau'_\$\tau'_{\overline{x}}\omega_\varepsilon = \sum_{l=0}^\infty\sum_{\overline{x}\in\Sigma^l}\sigma p\tau_\Sigma(1-p)^l\tau_{\overline{x}}\omega_\varepsilon = \sum_{l=0}^\infty p(1-p)^l = 1$. ■

Definition 24 *A terminating OOM $\mathcal{M}$ over the alphabet $\Sigma\cup\{\$\}$ and a SMA $\mathcal{A}$ over the alphabet Σ are* related*, if $f_\mathcal{M}(\overline{x}\$) = f_\mathcal{A}(\overline{x})$ for all $\overline{x}\in\Sigma^*$.*

Lemma 25 *If $\mathcal{A} = (\sigma, \{\tau_z\}, \omega_\varepsilon)$ is a minimal d-dimensional SMA, then $\tau_{\Sigma^*} = \sum_{k=0}^\infty \tau_\Sigma^k$ exists and is equal to $(I_d - \tau_\Sigma)^{-1}$, where $\tau_\Sigma = \sum_{z\in\Sigma}\tau_z$.*

Proof We will show that the spectral radius[1] $\rho(\tau_\Sigma)$ satisfies $\rho(\tau_\Sigma) < 1$, which implies the lemma. Assume $\rho(\tau_\Sigma) \geq 1$, i.e., there exists some $\lambda \in \mathbb{C}, |\lambda| \geq 1$ and $v \in \mathbb{C}^d$ such that $\tau_\Sigma v = \lambda v$. As $\mathcal{A}$ is minimal, we may find sequences $\overline{x}_j, \overline{x}_i \in \Sigma^*$ such that $\Pi = ((\sigma\tau_{\overline{x}_i})^\top)_{i\in I}^\top$ and $\Phi = (\tau_{\overline{x}_j}\omega_\varepsilon)_{j\in J}$ with $|I| = |J| = d$ are non-singular using Algorithm 1. Then $v = \Phi a$ for some $a \in \mathbb{C}^d$, and $\Pi\tau_\Sigma^k\Phi a = \lambda^k\Pi\Phi a$ for any $k \in \mathbb{N}$. Now the SMA property $f_\mathcal{A}(\Sigma^*) = \sum_{k=0}^\infty \sigma\tau_\Sigma^k\omega_\varepsilon = 1$ implies that $\Pi\tau_\Sigma^k\Phi \to 0$ as $k\to\infty$, while the right hand side $\lambda^k\Pi\Phi a$ does not (note $\Pi\Phi a \neq 0$), which is a contradiction. ■

1. For $A \in \mathbb{C}^{n\times n}$ with eigenvalues $\lambda_1, \ldots, \lambda_k$, the spectral radius is defined as $\rho(A) := \max_i |\lambda_i|$.

Proposition 26 *Let $\mathcal{A} = (\sigma, \{\tau_z\}, \omega_\varepsilon)$ be a minimal d-dimensional SMA. Then $\mathcal{M} = (\sigma', \{\tau'_z\}, \omega'_\varepsilon)$ is a related $(d+1)$-dimensional terminating OOM over the alphabet $\Sigma_\$ = \Sigma \cup \{\$\}$, if*

- $\sigma' = [\sigma \sum_{k=0}^{\infty} \tau_\Sigma^k, 1] = [\sigma(I_d - \tau_\Sigma)^{-1}, 1]$,
- $\tau'_z = \left[\begin{smallmatrix} \tau_z & 0 \\ 0 & 0 \end{smallmatrix}\right], \quad \tau'_\$ = \left[\begin{smallmatrix} 0 & 0 \\ \sigma & 1 \end{smallmatrix}\right]$, *and*
- $\omega'_\varepsilon = \left[\begin{smallmatrix} \omega_\varepsilon \\ 0 \end{smallmatrix}\right]$.

Proof We can simply check that for all $\overline{z} \in \Sigma_\$^*$

$$f_\mathcal{M}(\overline{z}) = \sigma' \tau'_{\overline{z}} \omega'_\varepsilon = \begin{cases} \sigma(\sum_{k=0}^{\infty} \tau_\Sigma^k) \tau_{\overline{z}} \omega_\varepsilon & \text{if } \overline{z} \in \Sigma^*, \\ \sigma \tau_{\overline{x}} \omega_\varepsilon & \text{if } \overline{z} = \overline{x}\$ \ldots \$ \text{ for some } \overline{x} \in \Sigma^*, \\ 0 & \text{otherwise.} \end{cases}$$

This implies $f_\mathcal{M} \geq 0$, $f_\mathcal{M}(\overline{x}\$) = f_\mathcal{A}(\overline{x})$ for all $\overline{x} \in \Sigma^*$ ($\mathcal{M}$ and $\mathcal{A}$ are related), as well as $f_\mathcal{M}(\Sigma^*\$) = f_\mathcal{A}(\Sigma^*) = 1$ ($\mathcal{M}$ is terminating if it is an OOM). Furthermore, $\sigma' \omega'_\varepsilon = f_\mathcal{A}(\Sigma^*) = 1$ and $\sigma' \tau'_{\Sigma_\$} = [\sigma \sum_{k=0}^{\infty} \tau_\Sigma^k \tau_\Sigma + \sigma, 1] = \sigma'$, which imply property (i) and (ii) for a stochastic process respectively. ■

Proposition 27 *Conversely, let $\mathcal{M} = (\sigma, \{\tau_z\}, \omega_\varepsilon)$ be a d-dimensional terminating OOM over the alphabet $\Sigma \cup \{\$\}$. Then $\mathcal{A} = (\sigma\tau_\$, \{\tau_z\}, \omega_\varepsilon)$ is a related d-dimensional SMA over the alphabet Σ.*

Proof Clearly, $f_\mathcal{A}(\overline{x}) = f_\mathcal{M}(\overline{x}\$) \geq 0$ for all $\overline{x} \in \Sigma^*$ and $f_\mathcal{A}(\Sigma^*) = f_\mathcal{M}(\Sigma^*\$) = 1$. ■

3.2.3 Historical Remarks

Note that our definition of OOMs given in Definition 20 differs slightly from the definition typically found in the literature.

First of all, the property (ii) for a stochastic process means that an OOM must satisfy $\sigma\tau_\Sigma\omega_{\overline{x}} = \sigma\omega_{\overline{x}}$ for all $\overline{x} \in \Sigma^*$, which implies (ii)' $\sigma\tau = \sigma$ if the OOM is minimal, but not in general. The property (ii)' is however often stated as part of the definition for OOMs. Our above Definition 20 is therefore slightly more relaxed than the standard definition in the case of non-minimal models, but this has no practical consequences.

Furthermore, for purely historical reasons, OOMs are sometimes required to satisfy $\sigma = (1, \ldots, 1)$, which is mainly an issue of normalization (cf. Proposition 13). However, this in turn has led to a more restrictive definition of interpretability for OOMs, since due to property (i) of stochastic processes, an OOM that satisfies $\sigma = (1, \ldots, 1)$ can only be interpretable with respect to sets Y_k, if $1 = \sigma\omega_\varepsilon = (1, \ldots, 1) \cdot [f_\mathcal{M}(Y_i)]_i^\top = \sum_k \sum_{\overline{y} \in Y_k} P(\overline{y})$. This is typically assured by requiring the sets Y_k to partition Σ^l for some l. One can relax this restriction on the sets Y_k for the definition of interpretability — as we have done in Definition 14 — if one is willing to drop the normalization requirement $\sigma = (1, \ldots, 1)$ as well.

Nevertheless, even though the normalization requirement $\sigma = (1, \ldots, 1)$ is superfluous, several of the OOM learning algorithms have been designed to yield OOMs normalized such that $\sigma = (1, \ldots, 1)$ — oftentimes unnecessarily complicating the algorithms — and some proofs have made use of this normalization as well. Later in Section 4 we present simplified and generalized versions of the EC and ES learning algorithms by removing this normalization restriction from the algorithms and proofs.

3.3 Predictive State Representations and Controlled Processes

Following the development of OOMs for stochastic processes, extensions to the case of controlled processes — stochastic processes that depend on an external input at each time step — were proposed by Jaeger (1998) as *input-output* OOMs, by Littman et al. (2001) as *predictive state representations* and as a further variant as *transformed PSRs* by Rosencrantz et al. (2004). All approaches are (in the linear case) equivalent and can be easily understood in the framework of linear SSs.

Definition 28 *A* (discrete-valued) controlled (stochastic) process *with input from Σ_I and output in Σ_O is a function $p : \Sigma^* \to [0, 1]$ that satisfies (i) $p(\varepsilon) = 1$ and (ii) $\forall \overline{x} \in \Sigma^*, a \in \Sigma_I : p(\overline{x}) = \sum_{o \in \Sigma_O} p(\overline{x}ao)$, where $\Sigma = (\Sigma_I \times \Sigma_O)$ and $ao = (a, o)$. We define $p(\overline{y}|\overline{x}) = p(\overline{x}\overline{y})/p(\overline{x})$ for $p(\overline{x}) \neq 0$ and zero otherwise. An* input-output OOM *(IO-OOM) is just a SS that models a controlled process.*

Note that the values of p are not probabilities. One may interpret $p(a_1 o_1 \ldots a_n o_n)$ as $P(o_1 \ldots o_n | a_1 \ldots a_n)$, i.e., as the conditional probability of observing the outputs $o_1 \ldots o_n$ given the inputs $a_1 \ldots a_n$. However, one must take care, as the sequence of inputs may depend on the observed outputs as well. This is explained in more detail in Section 4.1.

Definition 29 *Let p be a controlled process with* predictive states $\dot{\omega}_{\overline{h}}$ *defined as $\dot{\omega}_{\overline{h}} = [p(\overline{q}_1|\overline{h}), \ldots, p(\overline{q}_d|\overline{h})]^\top \in \mathbb{R}^d$ for $\overline{h} \in \Sigma^*$ and some fixed set of sequences $\overline{q}_i \in \Sigma^*$. If $\dot{\omega}_{\overline{h}}$ is a sufficient statistic for any history $\overline{h} \in \Sigma^*$, i.e., for every $\overline{x} \in \Sigma^*$ there is a function $m_{\overline{x}} : \mathbb{R}^d \to [0, 1]$ such that $p(\overline{x}|\overline{h}) = m_{\overline{x}}(\dot{\omega}_{\overline{h}})$ for all $\overline{h} \in \Sigma^*$, then the sequences $\{\overline{q}_1, \ldots, \overline{q}_d\}$ are called* core tests*, which together with the* initial state $\dot{\omega}_\varepsilon$ *and* projection functions $m_{\overline{x}}$ *form a d-dimensional* predictive state representation *(PSR) for p. If the projection functions are linear functionals (i.e., just row vectors in $\mathbb{R}^d$), then the PSR is called* linear*.*

Note that PSRs share the notion of "state" with OOMs in that the state consists of the information required to predict the future, but PSRs additionally require the entries of the state vectors $\dot{\omega}_{\overline{h}}$ to be "predictions" $p(\overline{q}_i|\overline{h})$ for the core tests $\overline{q}_i$. Such states are therefore called *predictive states*.

We will only consider linear PSRs for controlled processes here, and show that these are essentially SS for controlled processes (i.e., IO-OOMs) that are additionally interpretable with respect to singleton sets (core tests). Note that there has been some confusion about the precise relationship between PSRs and IO-OOMs, which we address in Sections 3.3.2 and 3.3.3 below.

Proposition 30 *Let a d-dimensional linear PSR consisting of core tests $\overline{q}_i$, projection functions $m_{\overline{x}}$ and an initial state $\dot{\omega}_\varepsilon$ for a controlled process p be given. Then an equivalent SS $\mathcal{M} = (\sigma, \{\tau_z\}, \omega_\varepsilon)$ is obtained by setting*

$$\omega_\varepsilon = \dot{\omega}_\varepsilon, \quad \tau_z = [(m_{z\overline{q}_1})^\top, \ldots, (m_{z\overline{q}_d})^\top]^\top \quad \text{and} \quad \sigma = \sum_{o \in \Sigma_O} m_{ao} \text{ for any } a \in \Sigma_I.$$

Furthermore, $\mathcal{M}$ will be interpretable w.r.t. the sets $\{\overline{q}_i\}$.

Proof First note that $\sigma\dot{\omega}_{\overline{x}} = \sum_{o\in\Sigma_O} m_{ao}\dot{\omega}_{\overline{x}} = \sum_{o\in\Sigma_O} p(ao|\overline{x}) = 1$ for all $\overline{x} \in \Sigma^*$ such that $p(\overline{x}) \neq 0$ because p is a controlled process. Next, we prove that (*) $\omega_{\overline{x}} = p(\overline{x})\dot{\omega}_{\overline{x}}$ and (**) $f_{\mathcal{M}}(\overline{x}) = p(\overline{x})$ by induction on the length l of $\overline{x}$:

- For $l = 0$ we have $\omega_\varepsilon = p(\varepsilon)\dot{\omega}_\varepsilon$ and $f_{\mathcal{M}}(\varepsilon) = \sigma\omega_\varepsilon = \sigma\dot{\omega}_\varepsilon = 1 = p(\varepsilon)$.
- Assume (*) and (**) are true for all $\overline{x} \in \Sigma^l$. Let $\overline{x}z \in \Sigma^{l+1}$. Then (*) $\omega_{\overline{x}z} = \tau_z\omega_{\overline{x}} = \tau_z\dot{\omega}_{\overline{x}}p(\overline{x}) = [p(z\overline{q}_i|\overline{x})]_i^\top p(\overline{x}) = [p(\overline{q}_i|\overline{x}z)]_i^\top p(z|\overline{x})p(\overline{x}) = \dot{\omega}_{\overline{x}z}p(\overline{x}z)$ and (**) $f_{\mathcal{M}}(\overline{x}z) = \sigma\omega_{\overline{x}z} = \sigma\dot{\omega}_{\overline{x}z}p(\overline{x}z) = p(\overline{x}z)$.

Note that property (*) says that $\omega_{\overline{x}} = p(\overline{x})\dot{\omega}_{\overline{x}} = [p(\overline{x}\overline{q}_1), \ldots, p(\overline{x}\overline{q}_d)]^\top$ for all $\overline{x}$, i.e., that $\mathcal{M}$ is interpretable w.r.t. the sets $\{\overline{q}_i\}$. ■

Proposition 31 *Conversely, let $\mathcal{M} = (\sigma, \{\tau_z\}, \omega_\varepsilon)$ be a SS for a controlled process p. Then an equivalent PSR is obtained by making the SS interpretable with respect to singleton sets $\{\overline{y}_i\}$ for appropriate sequences $\overline{y}_i \in \Sigma^*$ (e.g., using Algorithm 3). We can then use these as core tests for the PSR, and set $m_{\overline{x}} = \sigma\tau_{\overline{x}}$ for all $\overline{x} \in \Sigma^*$.*

Proof We assume that the SS has been made interpretable w.r.t. the sequences $\overline{y}_1, \ldots, \overline{y}_d$. Then the normalized states $\dot{\omega}_{\overline{h}} = \omega_{\overline{h}}/\sigma\omega_{\overline{h}}$ have the form $\dot{\omega}_{\overline{h}} = [p(\overline{y}_1|\overline{h}), \ldots, p(\overline{y}_d|\overline{h})]^\top$. Furthermore, for all $\overline{h} \in \Sigma^*$: $m_{\overline{x}}\dot{\omega}_{\overline{h}} = \sigma\tau_{\overline{x}}\dot{\omega}_{\overline{h}} = \sigma\tau_{\overline{x}}\tau_{\overline{h}}\omega_\varepsilon/\sigma\tau_{\overline{h}}\omega_\varepsilon = p(\overline{x}|\overline{h})$, as desired. ■

Corollary 32 *A linear PSR can be specified by the parameters $(\{m_{ao}\}, \{M_{ao}\}, \omega_\varepsilon^\top)$ for $ao \in \Sigma_I \times \Sigma_O$, where $M_{ao} = \tau_{ao}^\top$ and $m_{ao} = (\sigma\tau_{ao})^\top$, and defines a controlled process via*

$$p(a_1o_1 \cdots a_no_n) = \omega_\varepsilon^\top M_{a_1o_1} \cdots M_{a_{n-1}o_{n-1}} m_{a_no_n}.$$

This is the usual way of specifying a PSR.

Note that transformed PSRs (TPSRs) are just PSRs that model controlled processes in the form of Corollary 32 without any further requirements (i.e., without the requirement that the states need to be interpretable). These are readily converted to SSs by setting $\sigma = (\sum_{o\in\Sigma_O} m_{ao})^\top$ for any $a \in \Sigma_I$ and using the equations from the Corollary 32 otherwise. Note that this may not give equivalent models if the PSR does not model a controlled process.

3.3.1 Relation to Partially Observable Markov Decision Processes

Finally, we note how to convert POMDPs into SSs (which can then be further converted to PSRs by making the SS interpretable, as described above). A POMDP with d states $Q = \{s_1, \ldots, s_d\}$ for a controlled process with input alphabet Σ_I and output alphabet Σ_O consists of an initial belief state $b \in \mathbb{R}^d$ whose i-th element is the probability of the model starting in state s_i, a state transition matrix $T_a \in \mathbb{R}^{d \times d}$ for each action $a \in A$ such that the i,j-th entry of T_a is the probability of transitioning to state s_i from state s_j if action a is taken, and a vector $O_{ao} \in \mathbb{R}^d$ for each action-observation pair $ao \in (\Sigma_I \times \Sigma_O)$ whose i-th entry is the probability of observing o after arriving in state s_i by taking action a (Kaelbling et al., 1998).

Setting $O'_{ao} = \operatorname{diag}(O_{ao})$ we can summarize the belief-state update procedure for the POMDP concisely by stating that a POMDP models a controlled stochastic process p via the equation

$$p(a_1 o_1 \cdots a_n o_n) = (1, \ldots, 1)(O'_{a_n o_n} T_{a_n}) \cdots (O'_{a_1 o_1} T_{a_1}) b.$$

Clearly, setting $\sigma = (1, \ldots, 1)$, $\tau_{ao} = O'_{ao} T_a$ and $\omega_\varepsilon = b$ yields an equivalent SS.

3.3.2 IO-OOMs, Interpretable IO-OOMs, PSRs and TPSRs

We have shown above that IO-OOMs, PSRs and TPSRs are equivalent models in the sense that they model the same class of controlled processes and that they can be readily converted into one another. Furthermore, TPSRs are essentially IO-OOMs (except that the evaluation functional σ is replaced by the set $\{m_{ao}\}$ of evaluation functionals), while PSRs are TPSRs (and therefore essentially IO-OOMs) with predictive states, which corresponds to IO-OOMs being interpretable w.r.t. singleton sets (core tests). This is summarized in Table 1.

SSs for controlled processes with...	single evaluation functional σ	set of evaluation functionals $\{m_{ao}\}$
abstract, uninterpretable states	IO-OOMs	TPSRs
predictive states	IO-OOMs that are interpretable w.r.t. singleton sets	PSRs

Table 1: The differences between IO-OOMs, PSRs and TPSRs

Note that we have written "IO-OOMs that are interpretable w.r.t. singleton sets" instead of simply "interpretable IO-OOMs" for a reason. This is because interpretability was originally defined for IO-OOMs in a more restrictive way (cf. Section 3.3.3). It has been shown that not every IO-OOM has an equivalent "interpretable IO-OOM" (in the original sense) (Singh et al., 2004), i.e., that "interpretable IO-OOMs" are less general than IO-OOMs and PSRs. At the same time it was believed that some notion of interpretability would be crucial for the learnability of such models, which is however not the case, as we shall see in Section 4. Together, this has led to the false impression that PSRs are more general than IO-OOMs.

As the original notion of interpretability for IO-OOMs has turned out to be overly restrictive, we propose to employ the notion of interpretability that we have introduced here for SSs as the "correct" notion for IO-OOMs, and consider the original notion as deprecated.

3.3.3 Historical Remarks

The same remarks that we have made above in Section 3.2.3 for OOMs also apply to IO-OOMs. Namely, IO-OOMs were originally required to satisfy (ii)': $\forall a \in \Sigma_I : \sigma \sum_{o \in \Sigma_O} \tau_{ao} = \sigma$ instead of the the property (ii) for a controlled process. This is equivalent for minimal models, but slightly more restrictive in general. However, as every SS can be minimized, this has no practical consequences.

Furthermore, IO-OOMs were originally typically required to satisfy $\sigma = (1, \ldots, 1)$, which is again merely a matter of normalization. However, an IO-OOM that satisfies $\sigma = (1, \ldots, 1)$ can only be interpretable with respect to the sets Y_k, if $1 = \sigma\omega_\varepsilon = (1, \ldots, 1) \cdot [f_\mathcal{M}(Y_i)]_i^\top = \sum_k \sum_{\overline{y} \in Y_k} p(\overline{y})$. It turns out that this can be assured by requiring the sets Y_k to partition $\Sigma_O^l \times \{a_1\} \times \cdots \times \{a_l\}$ for some l and a fixed sequence $a_1 \ldots a_l$ of inputs called a *characterization frame*. This restriction on the choice of sets Y_k therefore became part of the original definition of interpretability for IO-OOMs.

Unfortunately, unlike the case for OOMs, the resulting original notion of interpretability for IO-OOMs has turned out to be a severe limitation (Singh et al., 2004).

However, one may use the more general notion of interpretability given in Definition 14 for IO-OOMs instead, if one is willing to drop the (unnecessary) normalization requirement $\sigma = (1, \ldots 1)$.

3.4 Extensions

In this section we have presented SMAs, OOMs and PSRs as versions of linear sequential systems — or more generally weighted finite automata — that model probabilistic languages, stochastic processes and controlled processes respectively, as is summarized in Figure 1. For completeness, we wish to briefly mention some extensions of these basic model types that have been studied, but which are beyond the scope of this paper.

First of all, various non-linear SSs exists. For instance, several versions of *quantum finite automata* have been studied (Kondacs and Watrous, 1997; Moore and Crutchfield, 2000). One form are SSs $(\sigma, \{\tau_x\}, \omega_\varepsilon \in \mathbb{C}P^d)$ where the operators τ_x are unitary and $\sigma(\tau_{\overline{x}}\omega_\varepsilon) = ||\pi\tau_{\overline{x}}\omega_\varepsilon||^2$ for some projection π and the Fubini-Study metric $||\cdot||$ (Moore and Crutchfield, 2000). A similar type of OOMs exist which are called *norm-OOMs*. These are SSs $(\sigma, \{\tau_x\}, \omega_\varepsilon \in \mathbb{R}^d)$ such that $\sum_{x \in \Sigma} \tau_x^\top \tau_x = I$ and $\sigma(\tau_{\overline{x}}\omega_\varepsilon) = ||\tau_x\omega_\varepsilon||^2$. Such norm-OOMs describe stochastic processes and can always be converted into an equivalent OOM (Zhao and Jaeger, 2010). Recently, *quadratic weighted automata* have been proposed by Bailly (2011), where a SS $\mathcal{M}$ is learnt for $\sqrt{f}$ and a product SS $\mathcal{M} \otimes \mathcal{M}$ is constructed that satisfies $f_{\mathcal{M}\otimes\mathcal{M}} = f_\mathcal{M}^2 \approx f$. All of these approaches avoid the "negative probabilities problem", where the estimated model $f_\mathcal{M}$ may violate the requirement $f_\mathcal{M} \geq 0$. Non-linear versions of PSRs have also been investigated, which have been shown to in some cases yield representations for deterministic dynamical systems that are exponentially smaller than a minimal OOM representation (Rudary and Singh, 2003).

Furthermore, OOMs and PSRs are models for discrete-valued stochastic (controlled) processes. Many real-world processes of interest are, however, continuous-valued. A continuous version of OOMs exists that extends semi-continuous HMMs (Jaeger, 2000a), and WFST have been similarly extended to allow for continuous inputs (Recasens and Quattoni, 2013). Multivariate continuous inputs and outputs are handled using features of observations by reduced-rank HMMs (Siddiqi et al., 2010). So called predictive linear Gaussian models (PLGs), which are based on PSRs, closely resemble linear dynamical system models (Rudary et al., 2005; Wingate and Singh, 2006a,b; Rudary and Singh, 2006, 2008) and are further generalized by exponential family PSRs (Wingate and Singh, 2008b,a). A generalization of OOMs using Hilbert space embeddings was introduced by Song et al. (2010). This has been further refined and extended to include features and can now be employed — among other things — for controlled processes and to planning in reinforcement learning tasks (Boots and Gordon, 2010; Boots et al., 2010, 2013).

4. Learning

In this section we present a general approach to learning SSs from data. We show how several of the learning algorithms that have been proposed for SMA, OOMs and PSRs can be understood in this framework, and that in fact many of the proposed learning algorithms are closely related.

We begin by establishing a result that lies at the heart of the learning algorithms, which was formulated by Kretzschmar (2001) for the case of OOMs. Assuming a function $f_{\mathcal{M}}$ can be described by some minimal SS $\mathcal{M}$, it allows us to reconstruct an equivalent SS $\mathcal{M}'$ from data given in the form of finitely many function values of $f_{\mathcal{M}}$ — as long as these are given exactly and we know the rank d of the underlying model $\mathcal{M}$. We will therefore refer to the Equations (2) as the *learning equations*.

Proposition 33 *For a minimal d-dimensional SS $\mathcal{M} = (\sigma, \{\tau_z\}, \omega_\varepsilon)$, let $\{\tau_{\overline{x}_j}\omega_\varepsilon \,|\, j \in J\}$ and $\{(\sigma\tau_{\overline{x}_i}^\top)^\top \,|\, i \in I\}$ be finite sets that span the state space W and the co-state space $\tilde{W}$ respectively. Define $F^{I,J} = [f_{\mathcal{M}}(\overline{x}_j\overline{x}_i)]_{(i,j)\in I\times J}$ and $F_z^{I,J} = [f_{\mathcal{M}}(\overline{x}_j z\overline{x}_i)]_{(i,j)\in I\times J}$. Furthermore, define $F^{I,0} = [f_{\mathcal{M}}(\overline{x}_i)]_{i\in I}$ and $F^{0,J} = [f_{\mathcal{M}}(\overline{x}_j)]_{j\in J}^\top$. Let $C \in \mathbb{R}^{d\times|I|}$ and $Q \in \mathbb{R}^{|J|\times d}$ be rank d matrices such that $CF^{I,J}Q$ is invertible. Then the SS $\mathcal{M}' = (\sigma', \{\tau_z'\}, \omega_\varepsilon')$ defined as follows is equivalent to $\mathcal{M}$:*

$$\begin{aligned} \sigma' &= F^{0,J}Q(CF^{I,J}Q)^{-1}, \\ \tau_z' &= CF_z^{I,J}Q(CF^{I,J}Q)^{-1}, \\ \omega_\varepsilon' &= CF^{I,0}. \end{aligned} \tag{2}$$

Furthermore, $CF^{I,J} = (\omega'_{\overline{x}_j})_{j\in J}$ and $CF_z^{I,J} = (\omega'_{\overline{x}_j z})_{j\in J}$, where $\omega'_{\overline{x}} = \tau'_{\overline{x}}\omega'_\varepsilon$ are states of the SS $\mathcal{M}'$.

Proof Let $\Pi = ((\sigma\tau_{\overline{x}_i})^\top)_{i\in I}^\top$, $\Phi = (\tau_{\overline{x}_j}\omega_\varepsilon)_{j\in J}$. Then $F^{I,J} = \Pi\Phi$, $F_z^{I,J} = \Pi\tau_z\Phi$, $F^{I,0} = \Pi\omega_\varepsilon$ and $F^{0,J} = \sigma\Phi$. We can then simply calculate $\tau_z' = C\Pi\tau_z\Phi Q(C\Pi\Phi Q)^{-1} = C\Pi\tau_z(C\Pi)^{-1}$, as well as $\omega_\varepsilon' = C\Pi\omega_\varepsilon$ and $\sigma' = \sigma\Phi Q(C\Pi\Phi Q)^{-1} = \sigma(C\Pi)^{-1}$. That is, we have shown that $\mathcal{M}' = \rho\mathcal{M}\rho^{-1}$ for the non-singular transformation $\rho = C\Pi$. Furthermore, $CF^{I,J} = C\Pi\Phi =$

$\rho\Phi = (\rho\tau_{\overline{x}_j}\omega_\varepsilon)_{j\in J} = (\tau'_{\overline{x}_j}\omega'_\varepsilon)_{j\in J}$, and analogously for $CF_z^{I,J}$. ■

The matrices C and Q that appear in the learning Equations (2) are indeed arbitrary (provided that $CF^{I,J}Q$ has the correct dimension d and full rank), as long as the function values $f_{\mathcal{M}}(\overline{x})$ are given exactly. However, if one only has access to estimates $\hat{f}(\overline{x})$, then the selection of C and Q plays a crucial role in obtaining good model estimates, as will be further discussed in Section 4.4.

Furthermore, note that we generally do not know a priori which sets of words to consider such that $\{\tau_{\overline{x}_j}\omega_\varepsilon \mid j \in J\}$ and $\{(\sigma\tau_{\overline{x}_i}^\top)^\top \mid i \in I\}$ span the state and co-state spaces W and $\tilde{W}$ of $\mathcal{M}$. Proposition 6 guarantees that it suffices to consider all words of length at most d, but the rank d of $\mathcal{M}$ is generally unknown as well. Selecting appropriate sets of words $\overline{x}_i$ and $\overline{x}_j$ and an appropriate model dimension d are therefore crucial and non-trivial steps in learning models from data.

We can turn the above Proposition 33 into a generic learning procedure for SSs:

Algorithm 4: General procedure for learning a SS from data

1. Obtain estimates $\hat{f}(\overline{x})$ of the function values $f(\overline{x})$ for words $\overline{x} \in \Sigma^*$.
2. Choose finite sets $\{\overline{x}_j \mid j \in J\}, \{\overline{x}_i \mid i \in I\} \subset \Sigma^*$, which we call sets of *indicative* and *characteristic* words respectively. Then assemble the estimates $\hat{f}(\overline{x})$ into estimates of the matrices $\hat{F}^{I,J}$, $\hat{F}_z^{I,J}$, $\hat{F}^{I,0}$ and $\hat{F}^{0,J}$.
3. Find a reasonable target dimension d for the model to be learnt.
4. Choose $C \in \mathbb{R}^{d\times|I|}$ and $Q \in \mathbb{R}^{|J|\times d}$ called the *characterizer* and *indicator*, such that $C\hat{F}^{I,J}Q$ is invertible.
5. Apply the learning Equations (2) to obtain a model estimate $\hat{\mathcal{M}}$.

At this point we should clarify what is meant here by learning a model from data. For general MA the goal is often to reconstruct an automaton from as few *membership queries* — obtaining the value $f(\overline{x})$ for some $\overline{x} \in \Sigma^*$ — and *equivalence queries* — proposing a function h and receiving a *counterexample* $\overline{x}$ such that $h(\overline{x}) \neq f(\overline{x})$ if $h \neq f$ — as possible. This is an extended version of the exact learning model of Angluin (1987). However, in the case of SMA, OOMs and PSRs, the external function represents a distribution. Therefore, in these cases it is usual to assume that we observe samples from this distribution and wish to estimate model parameters from the given samples such that the estimated model best describes the underlying distribution — "best" in a sense that depends on the context and the approach taken by a specific learning algorithm.

We should also mention one common problem when learning SMAs, OOMs and PSRs from data. Namely, even if the function $f_{\mathcal{M}}$ in question can be described by a SMA, OOM or PSR model $\mathcal{M}$, the learnt model $\hat{\mathcal{M}}$ will only be an approximation to $\mathcal{M}$ and will describe a function $f_{\hat{\mathcal{M}}}$ that may not satisfy the properties of a probabilistic language, stochastic process or controlled process, respectively, i.e., the learnt model $\hat{\mathcal{M}}$ may *not* be a SMA, OOM or PSR. What typically happens is that the learnt model $\hat{\mathcal{M}}$ will predict "negative probabilities" for certain sequences $\overline{x}$. Moreover, it is an undecidable problem whether a

given SS $\hat{\mathcal{M}}$ satisfies $f_{\hat{\mathcal{M}}} \geq 0$, and therefore, whether it is a SMA — a result that carries over to OOMs and PSRs as well (Wiewiora, 2008). In practice, there are three basic ways to deal with this "negative probabilities problem": First of all, one can resort to alternative models as described in Section 3.4 that preclude the problem by design. For the particular case of quadratic weighted automata the learning procedure presented here still applies (Bailly, 2011), but in general one will need alternative learning algorithms. Secondly, one may attempt to learn a restricted class of SS such as PFA, HMMs or POMDPs by enforcing additional constraints on the parameters of the SS. This can be achieved either by adding a set of convex constraints to a generalized version of the spectral learning method presented in Section 4.4.2 (Balle et al., 2012), or by an additional conversion step (Anandkumar et al., 2012), which however may fail. Finally, one may work with such an "invalid" SS model by employing a simple and effective heuristic as described by Jaeger et al. (2006b, Appendix J) to normalize all model predictions to fall into the desired range.

Finally, we will briefly remark on the runtime characteristics of the above learning procedure. Steps 1 and 2 can be accomplished in time $\mathcal{O}(N)$, where N is the size of the training data, for most strategies mentioned in Section 4.2 by employing a suffix tree or similar representation of the training data. For a given target dimension d, Step 4, when solved via the EC (Section 4.4.3) or spectral algorithms (Section 4.4.3), requires the $\mathcal{O}(d|I||J|)$ computation of a d-truncated singular value decomposition (SVD) of $\hat{F}^{I,J}$, while the ES algorithm (Section 4.4.4) requires $\mathcal{O}(d^2 l \max\{|I|, |J|\})$ operations to compute C, where l is the (generally very small) average length of characteristic and indicative words, and $\mathcal{O}(d|I||J|)$ operations to compute Q — per iteration (but one typically uses a constant number of iterations), which therefore amounts to a run-time of $\mathcal{O}(d|I||J|)$ as well. Solving the learning Equations (2) for Step 5 essentially requires the computation of the operators $\hat{\tau}_z$, which costs $\mathcal{O}(d|I||J||\Sigma|)$ operations. So for a known target dimension d, the above learning procedure typically requires $\mathcal{O}(N + d|I||J||\Sigma|)$ operations. Step 3 can be solved by computing a $d_{\max}$-truncated SVD of $\hat{F}^{I,J}$ for some upper bound $d_{\max} < \min\{|I|, |J|\}$ on the target dimension, which incurs a runtime costs of $\mathcal{O}(d_{\max}|I||J|)$, or by using cross-validation, which requires repeatedly performing, for various choices of d, Steps 4 and 5 as well as evaluations on test data of size T, which we assume to be constant, incurring a runtime cost of $\mathcal{O}(d \log(d)|I||J||\Sigma|)$, where d is the finally selected model dimension.

In the following, we will discuss the steps of the learning procedure in more detail.

4.1 Obtaining Estimates $\hat{f}(\overline{x})$

This step clearly depends on the context we are dealing with. Recall that in the context of SMA, the functions we are considering are distributions on words, while in the context of OOMs and PSRs they represent stochastic processes and controlled processes respectively. The following Remarks 34 to 36 summarize how to obtain these estimates in the different scenarios of probabilistic languages, stochastic processes and controlled processes, respectively.

Remark 34 *Let $f : \Sigma^* \to [0, 1]$ be a distribution on Σ^*, and let $S = (s_1, s_2, \ldots, s_N)$ be a collection of N samples from f. Then $\hat{f}(\overline{x}) = \frac{\#(\overline{x})}{N}$, where $\#(\overline{x})$ denotes the number of occurrences of $\overline{x}$ in the sample S, is a consistent estimator for $f(\overline{x})$.*

In the case of stochastic processes, one typically observes few (or even just one) long initial realization of the process. In this case it is still possible to obtain the desired estimates if the stochastic process is stationary and ergodic[2] by invoking the ergodic theorem and using time-averages as estimates. The same idea is commonly used in the case of controlled processes as well and called *suffix-history* method in the PSR community.

Remark 35 *Let $f : \Sigma^* \to [0,1]$ be a stationary and ergodic stochastic process, and let $\bar{s} = s_1 s_2 \dots s_N$ be a finite initial realization of length N from this process. Then*

$$\hat{f}(\overline{x}) = \frac{\#(\overline{x})}{N - |\overline{x}| + 1},$$

where $\#(\overline{x})$ denotes the number of occurrences of $\overline{x}$ in the sequence $\bar{s}$ is a consistent estimator for $f(\overline{x})$.

In the case of controlled processes the situation is more complicated. It is important to have a good understanding of the meaning of the value $f(\overline{x})$ when f is a controlled process and $\overline{x} = a_1 o_1 \dots a_n o_n \in (\Sigma_I \times \Sigma_O)^n$ is some input-output sequence. Intuitively, this is the probability of the system output $o_1 \dots o_n$ conditioned on the system input $a_1 \dots a_n$. This is sometimes written as $f(a_1 o_1 \dots a_n o_n) = P(o_1 \dots o_n \,|\, a_1 \dots a_n)$ even though this notation is misleading, as it suggests that $P(o_1 \dots o_n \,|\, a_1 \dots a_n) = \frac{P(a_1 o_1 \dots a_n o_n)}{P(a_1 \Sigma_O \dots a_n \Sigma_O)}$, which is false (Bowling et al., 2006). To clarify this, consider the stochastic process that is specified by the controlled process f *together with some system input specification.* This stochastic process is governed by probabilities of the form

$$P(a_1 o_1 \dots a_n o_n) = \prod_{k=1}^{n} P(o_k \,|\, a_1 o_1 \dots a_k) \cdot \prod_{k=1}^{n} P(a_k \,|\, a_1 o_1 \dots a_{k-1} o_{k-1}).$$

The second factor in the equation models the system input and is sometimes called the *input policy* π, while the first factor models the system output and is just the controlled process f. Therefore, for $\overline{x} = a_1 o_1 \dots a_n o_n$,

$$f(\overline{x}) = P(o_1 \dots o_n \,|\, a_1 \dots a_n) = \prod_{k=1}^{n} P(o_k \,|\, a_1 o_1 \dots a_k) = \frac{P(\overline{x})}{\pi(\overline{x})}. \tag{3}$$

Note that for the special case of a *blind* input policy π — one that does not depend on the observed output, i.e., that satisfies $P(a_k \,|\, a_1 o_1 \dots a_{k-1} o_{k-1}) = P(a_k \,|\, a_1 \dots a_{k-1})$ for all $\overline{x}$ — we in fact do have $\pi(\overline{x}) = P(a_1 \Sigma_O \dots a_n \Sigma_O)$.

From the above Equation (3), the following estimates are derived (Bowling et al., 2006):

Remark 36 *Let $f : \Sigma^* \to [0,1]$ be a controlled process, and let $\bar{s} = a_1 o_1 \dots a_N o_N$ be a finite initial sample from f according to some input policy π, such that the resulting stochastic process is stationary and ergodic. Then*

$$\hat{f}(\overline{x}) = \prod_{k=1}^{n} \frac{\#(a_1 o_1 \dots a_k o_k)}{\#(a_1 o_1 \dots a_k)}$$

2. A stationary ergodic process is a stochastic process where the statistical properties do not change with time (stationarity) and where these can be estimated as time-averages from a single long sample (ergodicity). For details, see for example the textbook by Gray (1988)

is a consistent estimator for $f(\overline{x})$. If the input policy π is known, then

$$\hat{f}(\overline{x}) = \frac{\#(\overline{x})}{N - |\overline{x}| + 1} \cdot \frac{1}{\pi(\overline{x})}$$

is also a consistent estimator which may be used instead. Again, $\#(\overline{x})$ denotes the number of occurrences of $\overline{x}$ in the sequence $\bar{s}$.

None of the above estimates exploits the rich structure of the matrix F. If required, some of the convex constraints that the matrix F must satisfy can be ensured by applying an additional normalization step to the estimated matrix $\hat{F}$, as done by McCracken and Bowling (2006). These convex constraints — including a convex relaxation of the rank constraint — may also be used to infer missing values if some entries $\hat{f}(\overline{x})$ cannot be obtained directly, which becomes relevant in the context of learning more general (e.g., non-stochastic) weighted automata (Balle and Mohri, 2012), or to infer sequence alignment when learning WFST from unaligned input-output sequences (Bailly et al., 2013).

4.2 Choosing Indicative and Characteristic Words

Choosing indicative and characteristic words $\{\overline{x}_j \,|\, j \in J\}, \{\overline{x}_i \,|\, i \in I\} \subset \Sigma^*$ is equivalent to selecting which columns J and rows I of the system matrix F to estimate. Clearly, it is only possible to obtain a correct estimate for f if I and J are selected such that $\text{rank}(F) = d = \text{rank}(F^{I,J})$. It is however unclear how to satisfy this if the true rank is unknown or even impossible if $\text{rank}(F) = \infty$ — as may often be the case for real-world examples. Determining an appropriate rank for the model will be discussed in the following section.

One approach is, however, to attempt to select minimal sets of indicative and characteristic words such that $\text{rank}(F) = \text{rank}(F^{I,J})$. Such minimal sets are called sets of *core histories* and *core tests* in the context of PSRs, and their selection is called the *discovery problem*. This problem is easily solved by Algorithm 1 once a (minimal) SS model for f is known. For the case where only function values of f are available, an iterative procedure has been proposed (James and Singh, 2004) that, starting with the empty words, adds in each iteration all length-one extensions of previously found core histories and tests, but retains only a minimal set needed to span $\hat{F}^{I,J}$. Since any noisy matrix is typically non-singular, some notion of numerical linear independence is used to decide which words to retain in each step. It is important to note that there exist simple examples of finite rank where this iterative procedure fails to deliver sets of core histories and tests (James and Singh, 2004), i.e., it does not in general solve the discovery problem. A similar algorithm called DEES has been proposed in the context of learning SMA (Denis et al., 2006). The algorithms for learning MA in the exact learning framework also work by finding a minimal set of indicative and characteristic words, but there it is assumed that the function f may be queried exactly, and furthermore equivalence queries are employed to find additional core tests and histories (Ohnishi et al., 1994; Bergadano and Varricchio, 1994; Beimel et al., 2000).

It is important to note that there is no requirement to find minimal or even small sets of indicative and characteristic words, i.e., one does not need to solve the discovery problem when learning SS models from data (and once a SS model has been learnt, the problem is easily solved by Algorithm 1). In fact, using small such sets means that less of the available

training data will enter the model estimation, i.e., the available data will be under-exploited. It is therefore desirable to use (much) larger sets of indicative and characteristic words than strictly needed.

An approach which is in some sense complementary is to use all sequences of a given length l. By Proposition 6 one can ensure $\text{rank}(F^{I,J}) = d$ by choosing $l \geq d$. However, this is highly impractical, since the size of $\hat{F}^{I,J}$ grows exponentially with l. Also, many of the estimates in $\hat{F}^{I,J}$ will be based on very few — if any — occurrences in the available training data. Nevertheless, choosing a length $l \ll d$ and utilizing as indicative as well as characteristic words all words of length l that occur at least once in the training data often gives good results (Zhao et al., 2009a).

A further approach is to select as indicative and characteristic words all those that actually occur in the data and therefore allow data-based estimates (Bailly et al., 2009). However, it is reasonable to disallow indicative (resp. characteristic) words that are suffixes (resp. prefixes) of some other indicative (resp. characteristic) word if they always occur at the same positions in the training data, as these would just lead to identical columns (resp. rows) in the estimated matrices that are based on the same parts of the training data (Jaeger et al., 2006b). Moreover, one may select only the words that occur most frequently in the data (Balle et al., 2014). These approaches yield a choice of indicative and characteristic words that is matched to the available training data and can be computed in time $\mathcal{O}(N)$ where N is the size of the training data by using a suffix tree or similar representation of the training data.

Finally, it is also possible to group words into sets of words (as is also done in Definition 14) that we call *events*, and to use indicative and characteristic events in place of words. This corresponds to adding the respective columns and rows in the matrices $\hat{F}^{I,J}, \hat{F}_z^{I,J}$, etc. and can be formally accomplished by a special selection of the indicator and characterizer matrices Q and C. Finding good indicative and characteristic events was the strategy adopted by early OOM learning algorithms (Jaeger, 2000b). A further generalization of this idea of considering events in place of words is proposed by Wingate et al. (2007). Using such events may carry an additional advantage if the estimation of $\hat{f}(Y)$ from the available data can be performed more efficiently or accurately than computing $\hat{f}(Y) = \sum_{\overline{x} \in Y} \hat{f}(\overline{x})$.

4.3 Determining the Model Rank

We should note that the goal of this step may be stated in two different ways. First of all, we may be interested in estimating the true rank of the external function f and use this as the model rank. On the other hand, we may rather be interested in choosing any model rank that allows for a good approximation of the external function f from the available data. These goals are related, as one can only hope to estimate an exact model if the model rank is at least $\text{rank}(f)$. However, they are not the same, and it depends on the context which approach is most appropriate. For instance, if it is known that the external function f must have a small finite rank, which may even carry some meaning, it may be desirable (and well-defined) to estimate this true rank from the data. On the other hand, when dealing with real-world systems of possibly infinite rank, and faced with generally limited training data, it may not even make sense to speak of the correct model rank. In such cases one will typically use the second approach, which is really an instance of the bias-variance dilemma.

4.3.1 Estimating the True Rank

For suitably chosen indicative and characteristic words, one can expect to have $\text{rank}(f) = \text{rank}(F^{I,J})$. However, since one only has access to an estimate $\hat{F}^{I,J}$ of this matrix, a typical approach is to determine what is known as the *numerical rank* (or *effective rank* or *pseudorank*). We give a brief description following Hansen (1998).

The *numerical ε-rank* r_ε of a matrix A may be defined as the smallest rank of any matrix that can be obtained from A by a small perturbation E of size at most ε:

$$r_\varepsilon(A) = \min_{||E|| \leq \varepsilon} \text{rank}(A + E).$$

In terms of the singular values $\sigma_1 \geq \cdots \geq \sigma_K$ of A this means that r_ε satisfies $\sigma_{r_\varepsilon} > \varepsilon \geq \sigma_{r_\varepsilon+1}$ if the size of the perturbation E is measured by the spectral norm $||\cdot||_2$, or alternatively that r_ε is the smallest k such that $\sum_{i=k+1}^{K} \sigma_i^2 \leq \varepsilon^2$ if the Frobenius norm $||\cdot||_F$ is used instead. Both criteria can be used to determine r_ε.

Assuming that A is only an estimate of an underlying matrix $\tilde{A}$, it makes sense to choose ε to be of the same order as the expected size of the error, i.e., $\varepsilon \approx E[||A - \tilde{A}||]$. The numerical rank of A is then $r_\varepsilon(A)$ for some reasonable choice of ε. Note that the notion of numerical rank makes sense if the errors on matrix entries of A are of comparable magnitudes and can be reasonably quantified, and if there is a significant *gap* between σ_{r_ε} and $\sigma_{r_\varepsilon+1}$. Otherwise, the numerical rank is somewhat arbitrary. It is furthermore important to note that the numerical rank measures how many dimensions can be significantly distinguished from noise. It is therefore only a lower bound for the true rank of the underlying matrix.

The main difficulty in determining the numerical rank of the matrix $\hat{F}^{I,J}$ therefore lies in finding a suitable ε. This may be approached by obtaining estimates for or bounds on the variances of the individual matrix entries (Jaeger, 1998; James and Singh, 2004), which may, however, differ widely across $\hat{F}^{I,J}$. These approaches will therefore lead to very conservative estimates of the rank. Still, these estimates will be consistent, i.e., will converge to the true rank in the limit of infinite training data.

Independent of such error estimates it may be reasonable to assume that there will be a relative "gap" between σ_{d+1} and σ_d in the singular value spectrum of $\hat{F}^{I,J}$ around the true rank $d = \text{rank}(F^{I,J})$. A recently proposed method searches for such a gap starting from σ_{r_ε}, where the numerical rank r_ε of $\hat{F}^{I,J}$ is used as a lower bound for the true rank (Bailly et al., 2009).

4.3.2 Finding a Suitable Model Rank

Intuitively speaking, the model rank should be chosen sufficiently large to be able to represent the complexity of the data, but not too large, as otherwise overfitting results.

One standard approach is to use cross-validation. For this, one needs to split the available data into training and test data. One then estimates models of various ranks from the training data and evaluates these on the test data, for instance by calculating the log likelihood of the test data under the models. Finally, one chooses the model rank that gives the best performance. Care must be taken when estimating models for controlled or stochastic processes from one long training sequence $\bar{s}$, as this sequence cannot be partitioned arbitrarily into training and test sets, and the distribution over future observations given a

history of observations at some time t may differ from the initial distribution. Additionally, performing cross-validation is computationally intense.

In comparison, the above methods based on calculating the numerical rank of $\hat{F}^{I,J}$ are elegant algebraic approaches to the problem. Recall that the numerical rank will reflect the number of dimensions present in the training data that can be distinguished from noise. It is therefore reasonable to postulate that the numerical rank of $\hat{F}^{I,J}$ might be a well-suited choice for the model dimension.

Interestingly, though, there is some evidence that at least the EC and spectral learning procedures described in the following section do not seem to suffer much from overfitting (Zhao et al., 2009a). In practical applications it may therefore be viable to simply pre-select a high model dimension.

Deeper insight into this crucial part of the learning procedure is unfortunately lacking. Further research into this question is therefore needed.

4.4 Selecting the Characterizer and Indicator

The effect of the characterizer C and indicator Q is to reduce the available data in $\hat{F}^{I,J}$, $\hat{F}_z^{I,J}$, $\hat{F}^{I,0}$ and $\hat{F}^{0,J}$ to a d-dimensional representation, where d is the chosen target dimension for the model to be learnt.

Assuming that $d = \text{rank}(F) = \text{rank}(CF^{I,J}Q)$, the matrices $CF^{I,J}Q, CF_z^{I,J}Q, CF^{I,0}$, and $F^{0,J}Q$ together contain the same information as F and are sufficient to reconstruct a SS model for f via the learning Equations (2). The requirement that $CF^{I,J}Q$ must have full rank d therefore ensures that no information is lost.

In fact — provided that $CF^{I,J}Q$ has full rank d — really any choice of characterizer and indicator *may* be used and will lead to a consistent model estimation, i.e., a correct model will be obtained in the limit of infinite training data. Hamilton et al. (2013) show that for certain dynamical systems a random choice of characterizer C does indeed work well.

However, in general the choice of characterizer C and indicator Q is central to achieving statistical efficiency, i.e., making efficient use of the available training data. This step lies at the heart of the learning procedure, and in fact much research — even if not explicitly stated — can be seen as optimizing this step of the learning algorithm.

4.4.1 By Selection / Grouping of Rows and Columns of $\hat{F}$

It is important to note that the choice of indicative and characteristic words discussed in Section 4.2 can be viewed equivalently as a special choice of characterizer and indicator. To see this, assume one could estimate the entire matrix $\hat{F}$ from data. Then any selection of rows I and columns J from $\hat{F}$ can be achieved by characterizer and indicator matrices C, Q of the form $C = C'C^I$ and $Q = Q^J Q'$, where C^I and Q^I are appropriate binary matrices with a single one entry in the corresponding columns or rows, and zeros otherwise, such that $C^I \hat{F} Q^J = \hat{F}^{I,J}$. This can easily be extended to account for groupings of words into events by allowing several one entries per column / row of C^I, Q^J respectively.

One advantage of this point of view is that this immediately justifies grouping of words into events, as suggested in Section 4.2. But more importantly, this highlights that choosing indicative and characteristic words as described in Section 4.2 is in fact a restricted approach to the more general problem of finding appropriate characterizer and indicator matrices. We

argue that a good choice of characterizer and indicator is the key to achieving high statistical efficiency of the learning procedure and that therefore the (pre-)selection of indicative and characteristic words should be guided by trying to retain as much information from the available training data as possible. In other words, the (pre-)selection of indicative and characteristic words in Section 4.2 is primarily a practical necessity that should rather be seen as discarding rows and columns from $\hat{F}$ that carry only little or no information.

4.4.2 Spectral Methods

Recall that the j-th columns of the matrices F and F_z correspond to the functions $f_{\overline{x}_j}$ and $f_{\overline{x}_j z}$, and that the operator τ_z of any minimal model $\mathcal{M}$ for f — regarded as a linear operator $\tilde{\tau}_z$ on the space $\mathcal{F}$ — satisfies $\tilde{\tau}_z(f_{\overline{x}_j}) = f_{\overline{x}_j z}$ (cf. Proposition 1). The matrix τ_z is just a representation of this operator with respect to some basis of $\mathcal{F}$. We can therefore regard the columns of F and F_z as argument-value pairs for the operator $\tilde{\tau}_z$, from which we can recover $\tilde{\tau}_z$. To obtain a matrix representation τ_z, we need to fix some basis for the column space $\mathcal{F}$, which corresponds to mapping the columns of F and F_z to $\mathbb{R}^d$ — this is accomplished by the characterizer C.

We are only given estimates $\hat{F}^{I,J}$ and $\hat{F}_z^{I,J}$. The idea of the spectral methods is to find an estimate of the column space $\hat{\mathcal{F}}$ by projecting the columns of $\hat{F}^{I,J}$ and $\hat{F}_z^{I,J}$ to a best rank d representation (best in the least squares sense). This is accomplished by the d-truncated SVD. We then estimate the matrices $\hat{\tau}_z$ via least squares linear regression from the so obtained argument-value pairs. Note that the column space $\mathcal{F}$ is already spanned by the columns of $F^{I,J}$ — if I and J are chosen appropriately — and we may therefore base the estimate of the principal subspace $\hat{\mathcal{F}}$ on the estimate $\hat{F}^{I,J}$ only. Formally, this means:

Algorithm 5: Spectral method for computing characterizer C and indicator Q

1 Compute $U_d S_d V_d^\top$, the d-truncated SVD of $\hat{F}^{I,J}$.
2 Set $C = U_d^\top$ and $Q = (C\hat{F}^{I,J})^\dagger = V_d S_d^\dagger$.

Note that $U_d S_d V_d^\top$ indeed gives the best rank d approximation to $\hat{F}^{I,J}$ with respect the Frobenius norm by the Eckart-Young theorem (Eckart and Young, 1936). However, the matrix $F^{I,J}_{\hat{\mathcal{M}}}$ reconstructed via the so learnt model $\hat{\mathcal{M}}$ — which will clearly have rank at most d — will in general *not* be a best rank d approximation to $\hat{F}^{I,J}$. This is due to the fact that constructing $F^{I,J}_{\hat{M}}$ from the model $\hat{\mathcal{M}}$ enforces additional structure. Interestingly, we have observed that the reconstructed matrix $F^{I,J}_{\hat{\mathcal{M}}}$ is often a better approximation to the *true* matrix $F^{I,J}$ than either of $\hat{F}^{I,J}$ and its best rank d approximation.

This spectral approach is often referred to as principal component analysis (PCA). However, PCA typically involves mean-centering the data first. PCA projects the data onto a d-dimensional *affine* subspace that contains the data mean, while here we know that the data $\hat{F}^{I,J}$ lie approximately on a true subspace (even though they do not have zero mean). Mean-centering the data is therefore inappropriate in this context — nevertheless, it it sometimes done anyway (Bailly et al., 2009). To avoid confusion, we refer to learning algorithms based on this idea simply as spectral learning algorithms (Rosencrantz et al., 2004; Hsu et al., 2009; Bailly et al., 2009; Siddiqi et al., 2010; Boots and Gordon, 2010; Bailly, 2011; Balle et al., 2011, 2014). Furthermore, an online version of this spectral

learning algorithm has been developed by Boots and Gordon (2011), whereas a modification that combines the subspace estimation step (determining the characterizer C) and linear regression step (solving the learning Equations 2) into a single optimization problem is given by Balle et al. (2012).

Clearly, these methods are motivated by trying to find a model $\hat{\mathcal{M}}$ of rank d such that its external function $f_{\hat{\mathcal{M}}}$ best approximates the estimated external function $\hat{f}$. To make this precise, one needs to define a distance measure on functions in $\mathbb{R}\langle\langle\Sigma\rangle\rangle$. In the case of stochastic languages the functions all lie in the Hilbert space $l_2(\Sigma^*)$ and the metric of this function space may be used. For stochastic processes, a natural choice may be the cross-entropy. This will be related to finding a maximum-likelihood estimate of model parameters from data. So far, none of these questions has been resolved. However, *sample complexity* results that fall into the probably approximately correct (PAC) learning framework (Valiant, 1984) are available for several spectral learning algorithms (Hsu et al., 2009; Bailly et al., 2009; Siddiqi et al., 2010; Bailly, 2011). These give bounds on the number or size N of samples that are required to obtain a model estimate $\mathcal{M}$ that is approximately correct (i.e., such that $|f_{\mathcal{M}} - f| < \varepsilon$ for a given ε and a specified distance measure) with probability at least $1 - \delta$ for a given δ. Typically, the required size N is shown to be polynomial in the PAC parameters $1/\epsilon$ and $1/\delta$, as well as other parameters that depend on f such as the alphabet size $|\Sigma|$ and the rank of f.

Finally, we mention a shortcoming of the spectral methods as they are commonly used. They implicitly assume that the variances of the estimates $\hat{f}(\overline{x}_j\overline{x}_i)$ are all of the same order. This, however, is clearly not the case, which suggests that replacing the SVD computation by a weighted low-rank matrix approximation (Markovsky and Huffel, 2007a) and the linear regression of the learning Equations (2) by weighted total least squares (Markovsky and Huffel, 2007b) may give better results, as long as weights that reflect the precision of the estimates $\hat{f}(\overline{x})$ can be estimated reliably from the available data. In fact, if the variances $\mathrm{Var}(\hat{f}(\overline{x}_j\overline{x}_i))$ can be estimated and — even approximately — factored as $\mathrm{Var}(\hat{f}(\overline{x}_j\overline{x}_i)) = v_j w_i > 0$, then this leads to a simple row and column weighted spectral learning method:

Algorithm 6: Row and column weighted spectral learning

1 Let $D_I = [\mathrm{diag}(w_i)_{i\in I}]^{-\frac{1}{2}}$ and $D_J = [\mathrm{diag}(v_j)_{j\in J}]^{-\frac{1}{2}}$ be suitable row and column weight matrices
2 Let $\tilde{F}^{I,J} = D_I \hat{F}^{I,J} D_J$ and $\tilde{F}_z^{I,J} = D_I \hat{F}_z^{I,J} D_J$
3 Let $\tilde{U}_d \tilde{S}_d \tilde{V}_d^\top$ be the d-truncated SVD of $\tilde{F}^{I,J}$
4 Let $C = \tilde{U}_d^\top D_I$ and $Q = D_J (C\tilde{F}^{I,J} D_J)^\dagger = D_J \tilde{V}_d \tilde{S}_d^\dagger$.

We mention this particular row and column weighted approach here, as it is simple, effective, and we will show that it is closely related to the ES approach described in Section 4.4.4.

4.4.3 The EC Algorithm

The error controlling (EC) approach selects characterizer and indicator matrices C and Q that minimize an error bound for the *relative approximation error* of the estimated model parameters (Zhao et al., 2009a). This algorithm was originally formulated for OOMs only, and made use of the normalization $\sigma = (1, \ldots, 1)$ that is often used in the context of OOMs.

This in turn imposed additional restrictions on the admissible selections of indicative and characteristic words. Here, we present a more general and yet simplified EC approach that eliminates these restrictions and applies to learning SMA, OOMs, IO-OOMs and PSRs alike.

To formalize this, first assume we have fixed C and Q, and derived estimated operators $\hat{\tau}_z$ and correct operators τ_z from the estimates $\hat{F}^{I,J}$, $\hat{F}_z^{I,J}$ and the correct matrices $F^{I,J}$, $F_z^{I,J}$ respectively using the learning Equations (2). Note that these depend on the choice of C and Q. To write things more concisely, denote the matrix obtained by stacking the τ_z operators by $\tau_* = [\tau_{z_1}; \ldots; \tau_{z_l}]$ (using MATLAB notation), where $\Sigma = \{z_1, \ldots, z_l\}$, and $\hat{\tau}_* = [\hat{\tau}_{z_1}; \ldots; \hat{\tau}_{z_l}]$. Similarly, construct the matrices $F_*^{I,J}$ and $\hat{F}_*^{I,J}$ by stacking the $F_z^{I,J}$ and $\hat{F}_z^{I,J}$ respectively.

Proposition 37 *For a given choice of C and Q, and using the above definitions, the estimate $\hat{\tau}_*$ has a* relative approximation error

$$\frac{\|\tau_* - \hat{\tau}_*\|_F}{\|\tau_*\|_F} \leq \kappa \left(\|F^{I,J} - \hat{F}^{I,J}\|_F + \frac{\sqrt{l}}{\rho(\tau_\Sigma)} \|F_*^{I,J} - \hat{F}_*^{I,J}\|_F \right),$$

where $\rho(\tau_\Sigma)$ is the spectral radius of the matrix τ_Σ, which is independent of the choice of C and Q, and $\kappa = \|C\|_F \|Q(C\hat{F}^{I,J}Q)^{-1}\|_F$.

This is a slightly improved and more general version of the central Proposition 3 presented in (Zhao et al., 2009a). For completeness, the proof is given in the appendix.

The EC algorithm then selects C, Q in such a way that the quantity κ is minimized, which is equivalent to the optimization problem

$$(C, Q) = \underset{(C,Q)}{\operatorname{argmin}} \{\|C\|_F \|Q\|_F : C\hat{F}^{I,J}Q = I_d\}, \tag{4}$$

since every (C, Q) that minimizes κ gives a solution (C, Q') to Equation (4) by substituting $Q' = Q(C\hat{F}^{I,J}Q)^{-1}$ and noting $(C\hat{F}^{I,J}Q') = I_d$. This optimization problem can be solved efficiently by the following iterative procedure (Zhao et al., 2009a):

Algorithm 7: The C, Q optimization resulting from the EC approach

initialize $C \in \mathbb{R}^{d \times |I|}$ randomly
repeat
  $Q = (C\hat{F}^{I,J})^\dagger, \quad C = (\hat{F}^{I,J}Q)^\dagger$
until *convergence of* $\|C\|_F \|Q\|_F$

Although not previously realized, this turns out to be related to a well-known EM-based algorithm for principal component analysis for which it is known that the rows of C (upon convergence) will span the space of the first d principle components of $\hat{F}^{I,J}$ (Roweis, 1998). We can use this relationship to gain the following insight.

Proposition 38 *Assuming the model rank d is chosen such that the singular values σ_i of $\hat{F}^{I,J}$ satisfy $\sigma_d > \sigma_{d+1}$, the EC algorithm as presented here and the spectral method presented in the previous section will lead to equivalent models.*

Proof Note that the condition $\sigma_d > \sigma_{d+1}$ merely says that $\operatorname{rank}(\hat{F}^{I,J}) \geq d$ and that the d-dimensional principal subspace of $\hat{F}^{I,J}$ is unique. Let C and $Q = (C\hat{F}^{I,J})^\dagger$ be the characterizer and indicator obtained by the spectral method, and let C' and $Q' = (C'\hat{F}^{I,J})^\dagger$ be the result of the above iterative procedure after convergence. Then the rows of C and C' will each span the same d-dimensional space (Roweis, 1998). This means that $C = \rho C'$ for some non-singular $\rho \in \mathbb{R}^{d\times d}$, and therefore $Q = (\rho C'\hat{F}^{I,J})^\dagger = (C'\hat{F}^{I,J})^\dagger\rho^{-1} = Q'\rho^{-1}$. By Proposition 12 the learning Equations (2) will result in equivalent models. ■

In fact, the above optimization problem can also be solved non-iteratively by a d-truncated SVD. This is a new result for which we give the full proof in the appendix:

Proposition 39 *Let $U_d S_d V_d^\top \approx \hat{F}^{I,J}$ be the d-truncated SVD of $\hat{F}^{I,J}$. Then $C^* = S_d^{-\frac{1}{2}} U_d^\top$ and $Q^* = (C^*\hat{F}^{I,J})^\dagger = V_d S_d^{-\frac{1}{2}}$ are a solution to the optimization problem in Equation (4) — provided a solution exists at all, i.e.,* $\operatorname{rank}(\hat{F}^{I,J}) \geq d$.

Clearly, this solution (C^*, Q^*) will again yield an equivalent model. Finally, we note that other versions of bounds on the relative approximation error than given in Proposition 37 may be considered instead, which can lead to choices of C and Q that give non-equivalent models. The performance of these seems to be comparable, though (Zhao et al., 2009b).

4.4.4 Efficiency Sharpening

The ES algorithm has previously been worked out only for the case of stationary stochastic processes and "traditional" OOMs where $\sigma = (1, \ldots, 1)$. Here we give an account of the ES principle that is more general than in the original work, and we establish connections to the spectral algorithms. The basic ES principle as we present it here may also be applied to learning SMA, IO-OOMs and PSRs from data. However, the concrete ES algorithm presented in Algorithm 8 makes use of several variance approximations and resulting simplifications that are only valid for the estimators from Remark 35 for the case of stationary stochastic processes.

The idea of the efficiency sharpening (ES) (Jaeger et al., 2006b) learning algorithm is to view the learning Equations (2) as a model estimator parameterized by C (and Q), and to select C such that the resulting estimator has minimum variance while still being consistent. Furthermore, this optimal choice of C is derived from knowledge of a model $\mathcal{M}$ for f, or in practice from a previous estimate thereof. To make this approach tractable, some simplifying assumptions are made.

First, a simplified version of the learning Equations (2) is used, where the indicator is taken to be $Q = (CF^{I,J})^\dagger$. This leads to operator estimates

$$\hat{\tau}_z = C\hat{F}_z^{I,J}(C\hat{F}^{I,J})^\dagger.$$

Jaeger et al. (2006b) now argue that due to the (pseudo)inversion, the variance of $\hat{\tau}_z$ is dominated by the variance of the factor $C\hat{F}^{I,J}$. The variance of a matrix is here taken w.r.t. the Frobenius norm. The ES algorithm therefore strives to find an admissible C such that the variance of $C\hat{F}^{I,J}$ is minimized — assuming knowledge of a model $\mathcal{M}$ for

f. A characterizer C is admissible if $CF^{I,J}Q$ is invertible. This is solved by the following proposition, which we state here in a more general form than in the original work (Jaeger et al., 2006b):

Proposition 40 *Let $\mathcal{M} = (\sigma, \{\tau_z\}, \omega_\varepsilon)$ be a d-dimensional minimal SS for a function $f : \Sigma^* \to \mathbb{R}$, and assume that $\hat{f}(\overline{x})$ are unbiased and uncorrelated estimators for all $\overline{x} \in \Sigma^*$. Define*

$$C^* = \Pi^\top D_I^2, \quad \text{where } \Pi^\top = ((\sigma\tau_{\overline{x}_i})^\top)_{i\in I}, \text{ and } D_I^2 = [\mathrm{diag}(\sum_{j\in J} \mathrm{Var}[\hat{f}(\overline{x}_j\overline{x}_i)])_{i\in I}]^\dagger.$$

Then $\mathrm{Var}[C\hat{F}^{I,J}]$ is minimized by the characterizer $C^* + 0$ among all characterizers of the form $C^* + G$ that satisfy $G\Pi = 0$.

The proof is given in the appendix, however, some explanatory remarks are in order. First of all, the assumptions that the estimates $\hat{f}(\overline{x})$ are unbiased and uncorrelated are reasonable, yet not strictly correct, meaning that the characterizer C^* will only approximate the theoretically optimal characterizer.

Next, we need a technical lemma to understand why it suffices to consider only characterizers of the form $(C^* + G)$ for some G satisfying $G\Pi = 0$:

Lemma 41 *If C^* has full row rank, then any admissible characterizer C can be written as $\rho(C^* + G)$ for some non-singular $\rho \in \mathbb{R}^{d\times d}$ and G such that $G\Pi = 0$.*

Proof Let C be some admissible characterizer. Then $C\Pi \in \mathbb{R}^{d\times d}$ must be invertible. Also, $C^*\Pi = (D_I\Pi)^\top(D_I\Pi)$ will be invertible if C^* has full row rank. Choosing $\rho = (C\Pi)(C^*\Pi)^{-1}$ and $G = \rho^{-1}(C - \rho C^*)$ we can easily verify that $C = \rho(C^* + G)$ and $G\Pi = 0$. ■

Note that the characterizers $C^* + G$ and $\rho(C^* + G)$ will lead to equivalent models via the learning Equations (2). Therefore, if the characterizer C^* is best among the class of characterizers $C^* + G$ where $G\Pi = 0$ then it is also the overall best choice.

Furthermore, the condition that C^* must have full row rank can be assured by (i) choosing indicative and characteristic sequences and the modeling dimension d accordingly, so that $d = \mathrm{rank}(\mathcal{M}) = \mathrm{rank}(F^{I,J}) = \mathrm{rank}(\Pi)$ and (ii) assuming that the variance of the estimators $\hat{f}(\overline{x})$ is non-zero, ensuring that D_I is invertible — which will typically be the case in practice.

Finally, to compute C^* via Proposition 40, we need to know the variances of the estimators $\hat{f}(\overline{x})$ occurring in D_I. Instead, we will replace D_I by an approximation that can be computed directly from the model $\mathcal{M}$. The approximation we present here is only valid for the case of stationary stochastic processes, but may be modified to cover the case of probabilistic languages as well.

Consider the estimators $\hat{f}(\overline{x})$ as in Remarks 34 and 35. It is reasonable to assume that the counts $\#(\overline{x})$ follow a binomial distribution, i.e., $\#(\overline{x}) \sim b_{N,p}$, where N is the length of the training sequence $\bar{s}$ and $p = f(\overline{x})$. This gives $\mathrm{Var}[\hat{f}(\overline{x})] = f(\overline{x})(1 - f(\overline{x}))/N$, which we may further approximate by $f(\overline{x})/N$, as in practice the values of $f(\overline{x})$ will typically be

small for most sequences $\overline{x}$. Also, the division by N is superfluous, as it cancels via the learning Equations (2). Using the approximation $\mathrm{Var}[\hat{f}(\overline{x})] \approx f(\overline{x})$, one can approximate

$$D_I^2 \approx \tilde{D}_I^2 := [\mathrm{diag}(\sum_{j \in J} f(\overline{x}_j \overline{x}_i))_{i \in I}]^\dagger = [\mathrm{diag}(\Pi \tau_{\overline{x}_J} \omega_\varepsilon)]^\dagger,$$

where $\tau_{\overline{x}_J} = \sum_{j \in J} \tau_{\overline{x}_j}$. The approximation

$$C^* \approx C^r := \Pi^\top \tilde{D}_I^2$$

is the characterizer that is actually used in the ES algorithm.

In the case of a stationary stochastic process and a choice of indicative words that partition Σ^l or $\Sigma^{\leq l}$ for some l one will have $\tau_{\overline{x}_J} \omega_\varepsilon = \omega_\varepsilon$, and therefore $\tilde{D}_I^2 = [\mathrm{diag}(\Pi \omega_\varepsilon)]^\dagger$. In this case, the columns $c_i = (\sigma \tau_{\overline{x}_i})^\top / \sigma \tau_{\overline{x}_i} \omega_\varepsilon$ of C^r can be seen as the normalized states $\omega^r_{\overline{x}_i{}^r} / \omega_\varepsilon^\top \omega^r_{\overline{x}_i{}^r}$ for the reversed words $\overline{x}_i{}^r$ under the *reversed* model $\mathcal{M}^\top = (\omega_\varepsilon^\top, \{\tau_z^\top\}, \sigma^\top)$, where $\omega^r_{\overline{x}_i{}^r} = \tau^\top_{(\overline{x}_i)_1} \cdots \tau^\top_{(\overline{x}_i)_k} \sigma^\top$. This is essentially the original version given by Jaeger et al. (2006b), and the reason why this characterizer was called the *reverse characterizer*. This make-up of C^r from states of the reversed process is also instrumental for the practical algorithms given by Jaeger et al. (2006b).

Additionally, the ES algorithm further exploits the interpretation of columns of $CF^{I,J}$ and $CF_z^{I,J}$ as model states $\omega_{\overline{x}_j}$ and $\omega_{\overline{x}_j z}$ as given in Proposition 33. These columns give argument-value pairs from which the operators τ_z can be deduced — as we have seen before. However, it is argued that in the face of estimates $\hat{F}^{I,J}$ and $\hat{F}_z^{I,J}$ the j-th columns should be weighted by $(\sum_{i \in I} \hat{f}(\overline{x}_j \overline{x}_i))^{-\frac{1}{2}}$ prior to performing linear regression to better reflect the weight of evidence that each column estimate is based on.

In practice a true model $\mathcal{M}$ is unknown. Therefore, the ES algorithm employs the following iterative procedure (again, our treatment here is more general than the original account by Jaeger et al. (2006b)):

Algorithm 8: The ES algorithm (for the case of stochastic processes)

1 Select some initial model estimate $\hat{\mathcal{M}}$ (e.g., via the learning Equations 2 using a random choice of C and Q).

repeat

2 Using the current model estimate $\hat{\mathcal{M}}$, compute $C = \hat{\Pi}^\top D_I^2$, where $\hat{\Pi}^\top = ((\hat{\sigma} \hat{\tau}_{\overline{x}_i})^\top)_{i \in I}$ and $D_I^2 = [\mathrm{diag}(\hat{\Pi} \sum_{j \in J} \hat{\tau}_{\overline{x}_j} \hat{\omega}_\varepsilon)_{i \in I}]^\dagger$.

3 Let $Q = D_J (C \hat{F}^{I,J} D_J)^\dagger$, where $D_J = [\mathrm{diag}(\sum_{i \in I} \hat{f}(\overline{x}_j \overline{x}_i))_{j \in J}]^{\dagger \frac{1}{2}}$.

4 Obtain a new model estimate $\hat{\mathcal{M}}$ via the learning Equations (2).

until *some fixed number of iterations, or some performance criteria of the estimated models stops increasing.*

Note that this procedure constructs a sequence of estimators along with a sequence of model estimates. The rationale of such ES algorithms is that the sequence of estimators increases in statistical efficiency, hence the name *efficiency sharpening* algorithms. The ES iterations come with no convergence guarantees. Nevertheless, this procedure has been found in practice to converge in very few iterations (3 – 5 typically suffice), and the results are of a similar quality as obtained by spectral algorithms (comparisons in Zhao et al., 2009a,b).

The ES algorithm is closely related to the row and column weighted spectral algorithm presented in Section 4.4.2. Precisely:

Proposition 42 *Assume $F^{I,J}$ of rank d is determined by some underlying minimal model $\mathcal{M} = (\omega_\varepsilon^\top, \{\tau_z^\top\}, \sigma^\top)$ of rank d and let $\Pi^\top = ((\sigma\tau_{\overline{x}_i})^\top)_{i\in I}$, $D_I = [\mathrm{diag}(\sum_{j\in J} f(\overline{x}_j\overline{x}_i))_{i\in I}]^{\dagger\frac{1}{2}}$ and $D_J = [\mathrm{diag}(\sum_{i\in I} f(\overline{x}_j\overline{x}_i))_{j\in J}]^{\dagger\frac{1}{2}}$. Let $C^r = \Pi^\top D_I^2$ be the reverse characterizer, and let $C' = \tilde{U}_d^\top D_I$ be the characterizer obtained by the weighted spectral method, where $\tilde{U}_d\tilde{S}_d\tilde{V}_d^\top$ is the d-truncated SVD of $D_I F^{I,J} D_J$. Then $C^r = \rho C'$ for some non-singular transformation ρ.*

Proof First, $\tilde{U}_d\tilde{S}_d\tilde{V}_d^\top = D_I F^{I,J} D_J$, since $F^{I,J}$ is assumed to have rank d. Now observe that $\tilde{U}_d\tilde{S}_d\tilde{V}_d^\top = D_I F^{I,J} D_J = D_I\Pi\Phi D_J$, where $\Phi = (\tau_{\overline{x}_j}\omega_\varepsilon)_{j\in J}$, and therefore the columns of $D_I F^{I,J}$, $\tilde{U}_d$ and $D_I\Pi$ all span $\mathrm{im}(D_I F^{I,J})$. So $C' = \tilde{U}_d^\top D_I$ and $C^r = (D_I\Pi)^\top D_I = \Pi^\top D_I^2$ have the same row space, and we can therefore find such a transformation ρ. ■

This means that the reverse characterizer C^r also gives a representation of the principal subspace of the weighted matrix $D_I F^{I,J}$. The main difference to the weighted spectral method described in Section 4.4.2 is that C^r is derived algebraically from an underlying model estimate, while the weighted spectral method estimates the principle subspace from the weighted data matrix $\hat{D}_I\hat{F}^{I,J}$ with weights $\hat{D}_I$ that also need to be determined from the data, e.g., $\hat{D}_I = [\mathrm{diag}(\sum_{j\in J} \hat{f}(\overline{x}_j\overline{x}_i))_{i\in I}]^{\dagger\frac{1}{2}}$.

5. Conclusion

We have shown that OOMs, PSRs and SMA are closely related instances of MA, and we have presented a unified learning framework for estimating such models from data that subsumes many of the existing learning algorithms. In presenting the learning framework, we have isolated the key design choices that need to be made to obtain a concrete learning algorithm. For each design choice we have surveyed the approaches that have been taken in the past and have tried to give some guidance.

We briefly summarize the choices that need to be made to obtain a concrete learning algorithm. First of all, estimates of the system matrices $\hat{F}^{I,J}$ and $\hat{F}_z^{I,J}$ must be obtained from the available training data. Individual entries may be estimated by the formulas given in Section 4.1. However, it is of much greater importance to decide which entries need to be estimated, that is, which rows I and columns J should be selected. This is discussed in Section 4.2. While many of the existing algorithms attempt to choose as few rows and columns to estimate as possible, we argue that this leads to poor statistical efficiency, and that the selection should ideally be matched to the available training data. Next, one must select a suitable model dimension d. This may be achieved by an algebraic criterion, as described in Section 4.3.1, or by cross-validation. It is also possible to treat this as a learning parameter that can be hand-tuned by the modeler. We note that it is generally neither necessary nor advisable to set the target dimension to the correct rank of the underlying system, as the optimal choice depends on the available training data. Finally, the estimated system matrices $\hat{F}^{I,J}$ and $\hat{F}_z^{I,J}$ need to be "compressed" to $d \times d$ matrices by suitable characterizer and indicator matrices C and Q. A good selection of C and Q is vital

to obtaining high statistical efficiency, and this is treated in detail in Section 4.4. We show that several of the proposed approaches to selecting C and Q can be seen as variations of a spectral learning algorithm presented in Section 4.4.2.

We conclude with a remark on implementing such a learning algorithm in practice. Clearly, the main limiting factor is the size of the matrices $\hat{F}^{I,J}$ and $\hat{F}_z^{I,J}$, as these may become very large. However, it is possible to obtain an efficient sparse representation of these matrices by employing a suffix tree representation of the training data (Zhao et al., 2009b,a; Jaeger et al., 2006b). Furthermore, if one uses the method described in Section 4.4.4 one can avoid evaluating these matrices explicitly and instead calculate $C\hat{F}^{I,J}$ and $C\hat{F}_z^{I,J}$ directly (Jaeger et al., 2006b).

Acknowledgements

We gratefully acknowledge the funding by the German Research Foundation (DFG) under the project JA 1210/5-1. We would also like to thank the anonymous reviewers for their constructive and very helpful comments.

Appendix

Proof [of Proposition 37](adapted from Zhao et al., 2009a) Let $C_* = \mathrm{diag}(C, \ldots, C)$ (l copies of C). Using the introduced notation the learning Equations (2) can be written concisely to obtain:

$$\begin{aligned}
\tau_* &= \left(C_*\hat{F}_*^{I,J}Q + C_*(F_*^{I,J} - \hat{F}_*^{I,J})Q\right)\left(C\hat{F}^{I,J}Q + C(F^{I,J} - \hat{F}^{I,J})Q\right)^{-1} \\
&= \left(C_*\hat{F}_*^{I,J}Q + C_*(F_*^{I,J} - \hat{F}_*^{I,J})Q\right)(C\hat{F}^{I,J}Q)^{-1}\left(I_d + C(F^{I,J} - \hat{F}^{I,J})Q(C\hat{F}^{I,J}Q)^{-1}\right)^{-1} \\
&= \left(\hat{\tau}_* + \left(C_*(F_*^{I,J} - \hat{F}_*^{I,J})Q\right)(C\hat{F}^{I,J}Q)^{-1}\right)\left(I_d + C(F^{I,J} - \hat{F}^{I,J})Q(C\hat{F}^{I,J}Q)^{-1}\right)^{-1},
\end{aligned}$$

which implies $\tau_* + \tau_* C(F^{I,J} - \hat{F}^{I,J})Q(C\hat{F}^{I,J}Q)^{-1} = \hat{\tau}_* + (C_*(F_*^{I,J} - \hat{F}_*^{I,J})Q)(C\hat{F}^{I,J}Q)^{-1}$. By rearranging, taking Frobenius norms and using the triangle inequality and submultiplicativity, we obtain

$$\|\tau_* - \hat{\tau}_*\|_F \leq \|C\|_F \|Q(C\hat{F}^{I,J}Q)^{-1}\|_F \left(\|\tau_*\|_F \|F^{I,J} - \hat{F}^{I,J}\|_F + \frac{\|C_*\|_F}{\|C\|_F}\|F_*^{I,J} - \hat{F}_*^{I,J}\|_F\right).$$

Now $\frac{\|C_*\|_F}{\|C\|_F} = \sqrt{l}$, and $\|\tau_*\|_F^2 = \sum_{z\in\Sigma}\|\tau_z\|_F^2 \geq \|\tau_\Sigma\|_F^2 \geq \rho(\tau_\Sigma)^2$, where $\tau_\Sigma = \sum_{z\in\Sigma}\tau_z$, and the result follows. ■

Note that in the original paper the inequality $\|\tau_*\|_F \geq \frac{1}{\sqrt{l}}$ was used instead, which depended on the columns of τ_* summing to 1. This was in turn insured by adding additional restrictions on the choice of characteristic words and characterizer C. These are now no longer needed.

Lemma 43 *Let $D = \mathrm{diag}(d_1, \ldots, d_n)$ and $S = \mathrm{diag}(s_1, \ldots, s_n)$ satisfying $d_1 \geq \cdots \geq d_n \geq 0$ and $0 \leq s_1 \leq \cdots \leq s_n$, and let U be an orthogonal $n \times n$ matrix, i.e., $U^\top U = UU^\top = I$. Then $\|DUS\|_F \geq \|DS\|_F$.*

Proof $\|DUS\|_F^2 = \sum_{i,j=1}^n (d_i u_{ij} s_j)^2$. Furthermore, $U^\top U = UU^\top = I_n$ implies that $(*)$ $\sum_{i=1}^n u_{i,j}^2 = 1$ for all j and $\sum_{j=1}^n u_{i,j}^2 = 1$ for all i. We will show the slightly stronger claim that $\|DUS\|_F^2 \geq \|DS\|_F^2$ for any matrix U satisfying $(*)$, which allows us to assume w.l.o.g. that $u_{i,j} \geq 0$ for all entries in U, since only the squared entries $u_{i,j}^2$ appear in the expressions for $\|DUS\|_F^2$ and $(*)$. So from now on we assume that U merely satisfies $(*)$ and that all entries in U are non-negative.

First note that if U is lower triangular, then $(*)$ implies that $U = I_n$: $\sum_{i=1}^n u_{i,n}^2 = 1$ implies that $u_{n,n}^2 = 1$ and $u_{i,n}^2 = 0$ for $i < n$. Then $\sum_{j=1}^n u_{n,j}^2 = 1$ implies that $u_{n,j}^2 = 0$ for $j < n$, since $u_{n,n}^2 = 1$. That is, $U = \left[\begin{smallmatrix} U_{n-1} & 0 \\ 0 & 1 \end{smallmatrix}\right]$, and the condition $(*)$ must therefore hold for U_{n-1} as well. By induction on n, $U = I_n$. In this case $\|DUS\|_F^2 = \|DS\|_F^2$.

So assume U is not lower triangular. Consider a row-wise ordering of matrix positions, i.e., define $\mathrm{ord}(i,j) = (i-1)n + j$, and let $(i',j') = \operatorname*{argmin}_{(i,j)} \{\mathrm{ord}(i,j) : j > i, u_{i,j} \neq 0\}$, i.e., i' is the first row of U to contain a non-zero element above the diagonal, and j' is the column index of the first such entry within the i'-th row. We call $\mathrm{ord}(i',j')$ the *order* of U, and say that a lower triangular matrix has infinite order.

Now consider the i'-th column of U. By the choice of i' we must have $\sum_{i=1}^{i'-1} u_{i,i'}^2 = 0$, and therefore $\sum_{i=i'+1}^n u_{i,i'}^2 = 1 - u_{i',i'}^2 = \sum_{j=1}^n u_{i',j}^2 - u_{i',i'}^2 \geq u_{i',j'}^2$. We can therefore find a vector v such that $v_i = 0$ for $i < i'$, $v_{i'} = -u_{i',j'}^2$, and $0 \leq v_i \leq u_{i,i'}^2$ as well as $\sum_{i=i'+1}^n v_i = u_{i',j'}^2$ for $i = i'+1, \ldots, n$. Let $U^2 = [u_{i,j}^2]_{i,j=1\ldots n}$ be the matrix of element-wise squares of entries in U, and let $\tilde{U}^2$ be obtained by subtracting the vector v from the i'-th column of U^2 and adding v to the j'-th column of U^2. Let $\tilde{U}$ be the matrix of element-wise square roots of entries in $\tilde{U}^2$.

We can easily check that all entries in $\tilde{U}^2$ are non-negative, so that this is well-defined. Also $\tilde{U}$ satisfies $(*)$, since $\sum_{i=1}^n v_i = 0$ by construction, and adding such a vector to one column of $\tilde{U}^2$ and subtracting from another does not change the row and column sums. Furthermore,

$$\begin{aligned}
\|DUS\|_F^2 - \|D\tilde{U}S\|_F^2 &= \sum_{i=1}^n \left(d_i^2 v_i s_{i'}^2 - d_i^2 v_i s_{j'}^2\right) \\
&= d_{i'}^2 v_{i'} (s_{i'}^2 - s_{j'}^2) + \sum_{i=i'+1}^n d_i^2 v_i (s_{i'}^2 - s_{j'}^2) \\
&= (s_{i'}^2 - s_{j'}^2) \left(d_{i'}^2 v_{i'} + \sum_{i=i'+1}^n d_i^2 v_i \right).
\end{aligned}$$

Now $s_{i'}^2 - s_{j'}^2 \leq 0$ since $j' > i'$, and $\sum_{i=i'+1}^n d_i^2 v_i \leq d_{i'}^2 \sum_{i=i'+1}^n v_i = d_{i'}^2 u_{i',j'}^2$, while $d_{i'}^2 v_{i'} = -d_{i'}^2 u_{i',j'}^2$, so $(d_{i'}^2 v_{i'} + \sum_{i=i'+1}^n d_i^2 v_i) \leq 0$. This shows that $\|DUS\|_F^2 \geq \|D\tilde{U}S\|_F^2$. And finally, the order of $\tilde{U}$ is larger than the order of U, as we have eliminated the non-zero element of lowest order above the diagonal in U, and in turn have introduced only non-zero elements above the diagonal of higher order (in rows below the i'-th), or none at all.

By iterating this construction we arrive at a lower triangular matrix U^* with non-negative entries that satisfies $(*)$ and $\|DUS\|_F^2 \geq \|DU^*S\|_F^2 = \|DS\|_F^2$. ■

Proof [Proof of Proposition 39] Assume $r = \text{rank}(\hat{F}^{I,J}) \geq d$ and let $USV^\top = \hat{F}^{I,J}$ be the full SVD of $\hat{F}^{I,J}$. We can simply verify that indeed $(C^*\hat{F}^{I,J})^\dagger = (S_d^{-\frac{1}{2}}U_d^\top USV^\top)^\dagger = (S_d^{-\frac{1}{2}}V_d^\top)^\dagger = V_d S_d^{-\frac{1}{2}}$, which implies that $C^*\hat{F}^{I,J}Q^* = C^*\hat{F}^{I,J}(C^*\hat{F}^{I,J})^\dagger = I_d$, as required. Furthermore, $\|C^*\|_F\|Q^*\|_F = \|S_d^{-\frac{1}{2}}\|_F^2 = \sum_{i=1}^d \sigma_i^{-1}$, where the σ_i are the singular values of $\hat{F}^{I,J}$, which are also the diagonal elements of S. We will show that this is indeed the minimum of $\|C\|_F\|Q\|_F$ subject to $C\hat{F}^{I,J}Q = I_d$.

Using the substitution $C = C'U^\top$ and $Q = VQ'$, we can see that minimizing $\|C\|_F\|Q\|_F$ subject to $C\hat{F}^{I,J}Q = I_d$ is equivalent to minimizing $\|C'\|_F\|Q'\|_F$ subject to $C'SQ' = I_d$ and that this will have the same minimal value. Let C'_r, Q'_r and S_r be truncated versions of C', Q' and S that consist of the first r columns, rows or rows and columns, respectively. Then minimizing $\|C'_r\|_F\|Q'_r\|_F$ subject to $C'_rS_rQ'_r = I_d$ is equivalent and has the same minimal value, because $C'SQ' = C'_rS_rQ'_r$ (since $\sigma_i = 0$ for $i > r$) and the additional columns in C' and rows in Q' are best set to zero.

Assume now that C'_r and $Q'_r = (C'_rS_r)^\dagger$ minimize $\|C'_r\|_F\|Q'_r\|_F$ subject to $C'_rS_rQ'_r = I_d$. We can select $Q'_r = (C'_rS_r)^\dagger$, as this minimizes $\|C'_r\|_F\|Q'_r\|_F$ subject to $C'_rS_rQ'_r = I_d$ for a given C'_r. It remains to show that $\|C'_r\|_F\|Q'_r\|_F \geq \|C^*\|_F\|Q^*\|_F = \sum_{i=1}^d \sigma_i^{-1}$.

Let $LDR^\top = C'_rS_r$ be the SVD of C'_rS_r. Then $C'_r = LDR^\top S_r^{-1}$, and $Q'_r = (C'_rS_r)^\dagger = RD^\dagger L^\top$. Let $d_1, \ldots, d_d$ be the diagonal elements of D and let D_r be the $r \times r$ matrix obtained by extending D with zero rows. Then

$$\|C'_r\|_F^2 = \|LDR^\top S_r^{-1}\|_F^2 = \|D_rR^\top S_r^{-1}\|_F^2 \overset{\text{Lemma 43}}{\geq} \|D_rS_r^{-1}\|_F^2 = \sum_{i=1}^d d_i^2\sigma_i^{-2},$$

$$\|Q'_r\|_F^2 = \|RD^\dagger L^\top\|_F^2 = \|D^\dagger\|_F^2 = \sum_{i=1}^d d_i^{-2}.$$

Multiplying these expressions and substituting $d_i^2 = a_i^2\sigma_i$, we obtain

$$\begin{aligned}
\|C'_r\|_F^2\|Q'_r\|_F^2 &= \left(\sum_{i=1}^d a_i^2\sigma_i^{-1}\right)\left(\sum_{i=1}^d a_i^{-2}\sigma_i^{-1}\right) \\
&= \sum_{i=1}^d \sigma_i^{-2} + \sum_{\substack{i,j=1 \\ i<j}}^d \left(\frac{a_i^2}{a_j^2} + \frac{a_j^2}{a_i^2}\right)\sigma_i^{-1}\sigma_j^{-1} \\
&= \sum_{i=1}^d \sigma_i^{-2} + \sum_{\substack{i,j=1 \\ i<j}}^d \left(\left(\frac{a_i}{a_j} - \frac{a_j}{a_i}\right)^2 + 2\right)\sigma_i^{-1}\sigma_j^{-1} \\
&\geq \left(\sum_{i=1}^d \sigma_i^{-1}\right)^2,
\end{aligned}$$

since this expression is clearly minimal when $a_i = 1$ for all i. So we can conclude that $\|C'_r\|_F\|Q'_r\|_F \geq \sum_{i=1}^d \sigma_i^{-1}$. Therefore, C^* and Q^* are in fact a minimal solution to the optimization problem (4). ■

Proof [of Proposition 40] First, we calculate:

$$\begin{aligned}
\mathrm{Var}[C\hat{F}^{I,J}] &\overset{(*)}{=} E\left[||C\hat{F}^{I,J} - CF^{I,J}||_F^2\right] \\
&= \sum_{j\in J}\sum_{k=1}^{d} E\left[\left(\sum_{i\in I} c_{ki}\hat{f}(\overline{x}_j\overline{x}_i) - \sum_{i\in I} c_{ki} f(\overline{x}_j\overline{x}_i)\right)^2\right] \\
&\overset{(*)}{=} \sum_{j\in J}\sum_{k=1}^{d} \mathrm{Var}\left[\sum_{i\in I} c_{ki}\hat{f}(\overline{x}_j\overline{x}_i)\right] \\
&\overset{(**)}{=} \sum_{j\in J}\sum_{k=1}^{d}\sum_{i\in I} c_{ki}^2 \mathrm{Var}[\hat{f}(\overline{x}_j\overline{x}_i)] \\
&= \sum_{i\in I} \|(C)_i\|_F^2 \sum_{j\in J} \mathrm{Var}[\hat{f}(\overline{x}_j\overline{x}_i)] = \sum_{i\in I} v_i \|(C)_i\|_F^2,
\end{aligned}$$

where $(C)_i$ is the i-th column of C, and $v_i = \sum_{j\in J} \mathrm{Var}[\hat{f}(\overline{x}_j\overline{x}_i)]$. Note that we have used unbiasedness in $(*)$ and uncorrelatedness in $(**)$.

Our goal is now to minimize $J(G) = \mathrm{Var}[(C^* + G)\hat{F}^{I,J}] = \sum_{i\in I} v_i ||(C^* + G)_i||_F^2$ subject to the constraints $h_{k,l}(G) = [G\Pi]_{k,l} = 0$ for $k, l = 1 \ldots d$. Note that if $v_i = 0$ for some i, then the i-th column of G does not influence the value of $J(G)$, and we may w.l.o.g. fix $(G)_i = 0$ and replace the equality constraints by $\tilde{h}_{k,l}(G) = [GDD^\dagger\Pi]_{k,l} = 0$, where $D = \mathrm{diag}[(v_i)_{i\in I}]$. This is a convex quadratic programming problem, therefore $G = 0$ will be a solution if and only if it satisfies the KKT conditions

$$\begin{aligned}
&\frac{\partial J}{\partial G}(G) + \sum_{k,l=1}^{d} \lambda_{k,l} \frac{\partial h_{k,l}}{\partial G}(G) = 0, \text{ and} \\
&\forall k, l = 1 \ldots d : \tilde{h}_{k,l}(G) = 0,
\end{aligned}$$

for some Lagrange multipliers $\lambda_{k,l} \in \mathbb{R}$. Clearly, the latter condition $\tilde{h}_{k,l}(G) = 0$ is satisfied for all k, l by $G = 0$. We can calculate $\sum_{k,l=1}^{d} \lambda_{k,l} \frac{\partial \tilde{h}_{k,l}}{\partial G}(G) = \lambda\Pi^\top D^\dagger D$, where $\lambda \in \mathbb{R}^{d\times d}, [\lambda]_{k,l} = \lambda_{k,l}$, as well as $\frac{\partial J}{\partial G}(G) = 2(C^* + G)D = 2(\Pi^\top D_I^2 + G)D$. The first condition is then satisfied by $G = 0$ with $\lambda = -2I$, since $\Pi^\top D_I^2 D = \Pi^\top D^\dagger D$ by definition of D_I. ■

References

Naoki Abe and Manfred K. Warmuth. On the computational complexity of approximating distributions by probabilistic automata. *Machine Learning*, 9:205–260, 1992.

Animashree Anandkumar, Daniel Hsu, and Sham M. Kakade. A method of moments for mixture models and hidden markov models. In Shie Mannor, Nathan Srebro, and Robert C. Williamson, editors, *Proceedings of the 25th Annual Conference on Learning*

Theory (COLT 2012), volume 23 of *JMLR Workshop & Conference Proceedings*, pages 33.1–33.34, 2012.

Dana Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, 1987.

Raphaël Bailly. Quadratic weighted automata: Spectral algorithm and likelihood maximization. In Chun-Nan Hsu and Wee Sun Lee, editors, *Proceedings of the 3rd Asian Conference on Machine Learning (ACML 2011)*, volume 20 of *JMLR Workshop & Conference Proceedings*, pages 147–163, 2011.

Raphaël Bailly, François Denis, and Liva Ralivola. Grammatical inference as a principal component analysis problem. In Andrea Pohoreckyj Danyluk, Léon Bottou, and Michael L. Littman, editors, *Proceedings of the 26th International Conference on Machine Learning (ICML 2009)*, volume 382 of *ACM Proceedings*, pages 33–40, 2009.

Raphael Bailly, Xavier Carreras, and Ariadna Quattoni. Unsupervised spectral learning of finite state transducers. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26 (NIPS 2013)*, pages 800–808. Curran Associates, Inc., 2013.

Borja Balle and Mehryar Mohri. Spectral learning of general weighted automata via constrained matrix completion. In Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25 (NIPS 2012)*, pages 2168–2176, 2012.

Borja Balle, Ariadna Quattoni, and Xavier Carreras. A spectral learning algorithm for finite state transducers. In Dimitrios Gunopulos, Thomas Hofmann, Donato Malerba, and Michalis Vazirgiannis, editors, *Machine Learning and Knowledge Discovery in Databases - European Conference (ECML/PKDD 2011), Proceedings, Part I*, volume 6911 of *Lecture Notes in Computer Science*, pages 156–171. Springer, 2011.

Borja Balle, Ariadna Quattoni, and Xavier Carreras. Local loss optimization in operator models: A new insight into spectral learning. In *Proceedings of the 29th International Conference on Machine Learning (ICML 2012)*. icml.cc / Omnipress, 2012.

Borja Balle, Xavier Carreras, Franco M. Luque, and Ariadna Quattoni. Spectral learning of weighted automata – a forward-backward perspective. *Machine Learning*, 96(1-2):33–63, 2014.

Amos Beimel, Francesco Bergadano, Nader H. Bshouty, Eyal Kushilevitz, and Stefano Varricchio. On the applications of multiplicity automata in learning. In *Proceedings of the 37th Annual Symposium on Foundations of Computer Science (FOCS 1996)*, pages 349–358. IEEE Computer Society, 1996.

Amos Beimel, Francesco Bergadano, Nader H. Bshouty, Eyal Kushilevitz, and Stefano Varricchio. Learning functions represented as multiplicity automata. *Journal of the ACM*, 47(3):506–530, 2000.

Francesco Bergadano and Stefano Varricchio. Learning behaviors of automata from multiplicity and equivalence queries. In Maurizio A. Bonuccelli, Pierluigi Crescenzi, and Rossella Petreschi, editors, *Proceedings of the 2nd Italian conference on Algorithms and Complexity (CIAC 1994)*, volume 778 of *Lecture Notes in Computer Science*, pages 54–62. Springer, 1994.

Francesco Bergadano, Dario Catalano, and Stefano Varricchio. Learning sat-k-DNF formulas from membership queries. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC 1996)*, pages 126–130. ACM, 1996.

Jean Berstel, Jr. and Christophe Reutenauer. *Rational Series and Their Languages*, volume 12 of *EATCS Monographs on Theoretical Computer Science*. Springer, 1988.

Byron Boots and Geoffrey J. Gordon. Predictive state temporal difference learning. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23 (NIPS 2010)*, pages 271–279. MIT Press, 2010.

Byron Boots and Geoffrey J. Gordon. An online spectral learning algorithm for partially observable nonlinear dynamical systems. In Wolfram Burgard and Dan Roth, editors, *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI 2011)*. AAAI Press, 2011.

Byron Boots, Sajid M. Siddiqi, and Geoffrey J. Gordon. Closing the learning-planning loop with predictive state representations. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, pages 1369–1370. IFAAMAS, 2010.

Byron Boots, Geoffrey J. Gordon, and Arthur Gretton. Hilbert space embeddings of predictive state representations. In *Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence (UAI 2013)*, pages 92–101. AUAI Press, 2013.

Michael Bowling, Peter McCracken, Michael James, James Neufeld, and Dana F. Wilkinson. Learning predictive state representations using non-blind policies. In William W. Cohen and Andrew Moore, editors, *Proceedings of the 23rd International Conference on Machine Learning (ICML 2006)*, volume 148 of *ACM Proceedings*, pages 129–136, 2006.

Jack W. Carlyle and Azaria Paz. Realizations by stochastic finite automata. *Journal of Computer and System Sciences*, 5(1):26–40, 1971.

Corinna Cortes and Mehryar Mohri. Context-free recognition with weighted automata. *Grammars*, 3(2/3):133–150, 2000.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood estimation from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39 (1):1–38, 1977.

François Denis and Yann Esposito. Learning classes of probabilistic automata. In *Proceedings of the 17th Annual Conference on Learning Theory (COLT 2004)*, volume 3120 of *Lecture Notes in Computer Science*, pages 124–139. Springer, 2004.

François Denis and Yann Esposito. On rational stochastic languages. *Fundamenta Informaticae*, 86(1):41–77, 2008.

François Denis, Yann Esposito, and Amaury Habrard. Learning rational stochastic languages. In Gábor Lugosi and Hans-Ulrich Simon, editors, *Proceedings of the 19th Annual Conference on Learning Theory (COLT 2006)*, volume 4005 of *Lecture Notes in Computer Science*, pages 274–288. Springer, 2006.

Manfred Droste, Werner Kuich, and Heiko Vogler. *Handbook of Weighted Automata*. Springer, 2009.

Pierre Dupont, François Denis, and Yann Esposito. Links between probabilistic automata and hidden Markov models: probability distributions, learning models and induction algorithms. *Pattern Recognition*, 38(9):1349–1371, 2005.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

Michel Fliess. Matrices de Hankel. *Journal de Mathématiques Pures et Appliquées*, 53: 197–222, 1974.

Edgar J. Gilbert. On the identifiability problem for functions of finite Markov chains. *The Annals of Mathematical Statistics*, 30(3):688–697, 1959.

Robert M. Gray. *Probability, Random Processes, and Ergodic Properties*. Springer, 1988.

William L. Hamilton, Mahdi M. Fard, and Joelle Pineau. Modelling sparse dynamical systems with compressed predictive state representations. In Sanjoy Dasgupta and David Mcallester, editors, *Proceedings of the 30th International Conference on Machine Learning (ICML 2013)*, volume 28 of *JMLR Workshop & Conference Proceedings*, pages 178–186, 2013.

Per Christian Hansen. *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1998.

Alex Heller. On stochastic processes derived from Markov chains. *The Annals of Mathematical Statistics*, 36(4):1286–1291, 1965.

Daniel Hsu, Sham M. Kakade, and Tong Zhang. A spectral algorithm for learning hidden Markov models. In *Proceedings of the 22nd Annual Conference on Learning Theory (COLT 2009)*, 2009.

Hisashi Ito. *An Algebraic Study of Discrete Stochastic Systems*. Unpublished doctoral dissertation, University of Tokyo, Bunkyo-ku, Tokyo, 1992.

Hisashi Ito, Shun ichi Amari, and Kingo Kobayashi. Identifiability of hidden Markov information sources and their minimum degrees of freedom. *IEEE Transactions on Information Theory*, 38(2):324–333, 1992.

Herbert Jaeger. Observable operator models and conditioned continuation representations. Arbeitspapiere der GMD 1043, GMD Forschungszentrum Informationstechnik, Sankt Augustin, Germany, 1997.

Herbert Jaeger. Discrete-time, discrete-valued observable operator models: a tutorial. Technical Report 42, GMD Forschungszentrum Informationstechnik, Sankt Augustin, Germany, 1998.

Herbert Jaeger. Modeling and learning continuous-valued stochastic processes with OOMs. GMD Report 102, GMD Forschungszentrum Informationstechnik, Sankt Augustin, Germany, 2000a.

Herbert Jaeger. Observable operator models for discrete stochastic time series. *Neural Computation*, 12(6):1371–1398, 2000b.

Herbert Jaeger, MingJie Zhao, and Andreas Kolling. Efficient estimation of ooms. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18 (NIPS 2005)*, pages 555–562. MIT Press, 2006a.

Herbert Jaeger, MingJie Zhao, Klaus Kretzschmar, Tobias Oberstein, Dan Popovici, and Andreas Kolling. Learning observable operator models via the ES algorithm. In Simon Haykin, José C. Príncipe, Terrence J. Sejnowski, and John McWhirter, editors, *New Directions in Statistical Signal Processing: From Systems to Brains*, Neural Information Processing, chapter 14, pages 417–464. MIT Press, Cambridge, MA, USA, 2006b.

Michael R. James and Satinder P. Singh. Learning and discovery of predictive state representations in dynamical systems with reset. In Carla E. Brodley, editor, *Proceedings of the 21st International Conference on Machine Learning (ICML 2004)*, volume 69 of *ACM Proceedings*, pages 53–60, 2004.

Michael R. James and Satinder P. Singh. Planning in models that combine memory with predictive representations of state. In Manuela M. Veloso and Subbarao Kambhampati, editors, *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI 2005)*, pages 987–992. AAAI Press, 2005.

Michael R. James, Satinder Singh, and Michael L. Littman. Planning with predictive state representations. In *Proceedings of the 3rd International Conference on Machine Learning and Applications (ICMLA 2004)*, pages 304–311. IEEE Computer Society, 2004.

Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1-2):99–134, 1998.

Attila Kondacs and John Watrous. On the power of quantum finite state automata. In *Proceedings of the 37th Annual Symposium on Foundations of Computer Science (FOCS 1996)*, pages 66–75. IEEE Computer Society, 1997.

Klaus Kretzschmar. Learning symbol sequences with observable operator models. GMD Report 161, GMD Forschungszentrum Informationstechnik, Sankt Augustin, Germany, 2001.

Michael L. Littman, Richard S. Sutton, and Satinder P. Singh. Predictive representations of state. In Thomas G. Dietterich, Suzanna Becker, and Zoubin Ghahramani, editors, *Advances in Neural Information Processing Systems 14 (NIPS 2001)*, pages 1555–1561. MIT Press, 2001.

Ivan Markovsky and Sabine Van Huffel. Left vs right representations for solving weighted low-rank approximation problems. *Linear Algebra and its Applications*, 422(2-3):540–552, 2007a.

Ivan Markovsky and Sabine Van Huffel. Overview of total least-squares methods. *Signal Processing*, 87(10):2283–2302, 2007b.

Peter McCracken and Michael H. Bowling. Online discovery and learning of predictive state representations. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18 (NIPS 2005)*. MIT Press, 2006.

Mehryar Mohri, Fernando Pereira, and Michael Riley. Weighted finite-state transducers in speech recognition. *Computer Speech & Language*, 16(1):69–88, 2002.

Cristopher Moore and James P. Crutchfield. Quantum automata and quantum grammars. *Theoretical Computer Science*, 237(1-2):275–306, 2000.

Hiroyuki Ohnishi, Hiroyuki Seki, and Tadao Kasami. A polynomial time learning algorithm for recognizable series. *IEICE Transactions on Information and Systems*, E77-D(10): 1077–1085, 1994.

Lawrence R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

Adrià Recasens and Ariadna Quattoni. Spectral learning of sequence taggers over continuous sequences. In Hendrik Blockeel, Kristian Kersting, Siegfried Nijssen, and Filip Zelezný, editors, *Machine Learning and Knowledge Discovery in Databases - European Conference (ECML/PKDD 2013), Proceedings, Part I*, volume 8188 of *Lecture Notes in Computer Science*, pages 289–304. Springer, 2013.

Matthew Rosencrantz, Geoffrey J. Gordon, and Sebastian Thrun. Learning low dimensional predictive representations. In Carla E. Brodley, editor, *Proceedings of the 21st International Conference on Machine Learning (ICML 2004)*, volume 69 of *ACM Proceedings*, pages 695–702, 2004.

Sam Roweis. EM algorithms for PCA and SPCA. In Michael I. Jordan, Michael J. Kearns, and Sara A. Solla, editors, *Advances in Neural Information Processing Systems 10 (NIPS 1997)*, pages 626–632. MIT Press, 1998.

Matthew Rudary and Satinder P. Singh. Predictive linear-Gaussian models of controlled stochastic dynamical systems. In William W. Cohen and Andrew Moore, editors, *Proceedings of the 23rd International Conference on Machine Learning (ICML 2006)*, volume 148 of *ACM Proceedings*, pages 777–784, 2006.

Matthew Rudary and Satinder P. Singh. Predictive linear-Gaussian models of stochastic dynamical systems with vector-value actions and observations. In *Proceedings of the 10th International Symposium on Artificial Intelligence and Mathematics (ISAIM 2008)*, 2008.

Matthew Rudary, Satinder P. Singh, and David Wingate. Predictive linear-Gaussian models of stochastic dynamical systems. In Fahiem Bacchus and Tommi Jaakkola, editors, *Proceedings of the 21st Conference in Uncertainty in Artificial Intelligence (UAI 2005)*, pages 501–508. AUAI Press, 2005.

Matthew R. Rudary and Satinder Singh. A nonlinear predictive state representation. In S. Thrun S. Becker and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15 (NIPS 2002)*, pages 855–862. MIT Press, 2003.

Arto Salomaa and Matti Soittola. *Automata-Theoretic Aspects of Formal Power Series*. Texts and Monographs in Computer Science. Springer, 1978.

Marcel Paul Schützenberger. On the definition of a family of automata. *Information and Control*, 4(2-3):245–270, 1961.

Sajid M. Siddiqi, Byron Boots, and Geoffrey J. Gordon. Reduced-rank hidden markov models. In Yee Whye Teh and D. Mike Titterington, editors, *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS 2010)*, volume 9 of *JMLR Workshop & Conference Proceedings*, pages 741–748, 2010.

Satinder Singh, Michael R. James, and Matthew R. Rudary. Predictive state representations: A new theory for modeling dynamical systems. In Joseph Halpern, editor, *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence (UAI 2004)*, pages 512–519. AUAI Press, 2004.

Le Song, Byron Boots, Sajid M. Siddiqi, Geoffrey J. Gordon, and Alex J. Smola. Hilbert space embeddings of hidden Markov models. In Johannes Fürnkranz and Thorsten Joachims, editors, *Proceedings of the 27th International Conference on Machine Learning (ICML 2010)*, pages 991–998. Omnipress, 2010.

Daniel R. Upper. *Theory and Algorithms for Hidden Markov Models and Generalized Hidden Markov Models*. PhD thesis, University of California at Berkeley, 1997.

Leslie G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

Eric W. Wiewiora. *Modeling Probability Distributions with Predictive State Representations*. PhD thesis, University of California, San Diego, 2008.

David Wingate and Satinder P. Singh. Kernel predictive linear Gaussian models for nonlinear stochastic dynamical systems. In William W. Cohen and Andrew Moore, editors, *Proceedings of the 23rd International Conference on Machine Learning (ICML 2006)*, volume 148 of *ACM Proceedings*, pages 1017–1024, 2006a.

David Wingate and Satinder P. Singh. Mixtures of predictive linear Gaussian models for nonlinear, stochastic dynamical systems. In Anthony Cohn, editor, *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI 2006)*. AAAI Press, 2006b.

David Wingate and Satinder P. Singh. Exponential family predictive representations of state. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20 (NIPS 2007)*, pages 1617–1624. MIT Press, 2008a.

David Wingate and Satinder P. Singh. Efficiently learning linear-linear exponential family predictive representations of state. In William W. Cohen, Andrew McCallum, and Sam T. Roweis, editors, *Proceedings of the 25th International Conference on Machine Learning (ICML 2008)*, volume 307 of *ACM Proceedings*, pages 1176–1183, 2008b.

David Wingate, Vishal Soni, Britton Wolfe, and Satinder P. Singh. Relational knowledge with predictive state representations. In Manuela M. Veloso, editor, *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007)*, pages 2035–2040. AAAI Press, 2007.

Britton Wolfe and Satinder P. Singh. Predictive state representations with options. In William W. Cohen and Andrew Moore, editors, *Proceedings of the 23rd International Conference on Machine Learning (ICML 2006)*, volume 148 of *ACM Proceedings*, pages 1025–1032, 2006.

Lotfi Asker Zadeh. The concept of system, aggregate, and state in system theory. In Lotfi Asker Zadeh and Elijah Polak, editors, *System Theory*, volume 8 of *Inter-University Electronics Series*, pages 3–42. McGraw-Hill, New York, 1969.

MingJie Zhao and Herbert Jaeger. Norm observable operator models. *Neural Computation*, 22(7):1927–1959, 2010.

MingJie Zhao, Herbert Jaeger, and Michael Thon. A bound on modeling error in observable operator models and an associated learning algorithm. *Neural Computation*, 21(9):2687–2712, 2009a.

MingJie Zhao, Herbert Jaeger, and Michael Thon. Making the error-controlling algorithm of observable operator models constructive. *Neural Computation*, 21(12):3460–3486, 2009b.

Journal of Machine Learning Research 16 (2015) 149-153 Submitted 6/14; Revised 9/14; Published 1/15

SAMOA: Scalable Advanced Massive Online Analysis

Gianmarco De Francisci Morales GDFM@APACHE.ORG
Albert Bifet ABIFET@YAHOO.COM
Yahoo Labs
Av. Diagonal 177, 8th floor, 08018, Barcelona, Spain

Editor: Geoff Holmes

Abstract

SAMOA (SCALABLE ADVANCED MASSIVE ONLINE ANALYSIS) is a platform for mining big data streams. It provides a collection of distributed streaming algorithms for the most common data mining and machine learning tasks such as classification, clustering, and regression, as well as programming abstractions to develop new algorithms. It features a pluggable architecture that allows it to run on several distributed stream processing engines such as Storm, S4, and Samza. SAMOA is written in Java, is open source, and is available at `http://samoa-project.net` under the Apache Software License version 2.0.

Keywords: data streams, distributed systems, classification, clustering, regression, toolbox, machine learning

1. Introduction

Big data is "data whose characteristics forces us to look beyond the traditional methods that are prevalent at the time" (Jacobs, 2009). Currently, there are two main ways to deal with big data: streaming algorithms and distributed computing (e.g., MapReduce). SAMOA aims at satisfying the future needs for big data stream mining by combining the two approaches in a single platform under an open source umbrella.

Data mining and machine learning are well established techniques among web companies and startups. Spam detection, personalization, and recommendation are just a few of the applications made possible by mining the huge quantity of data available nowadays.

The usual pipeline for mining and modeling data (what "data scientists" do) involves taking a sample from production data, cleaning and preprocessing it to make it amenable to modeling, training a model for the task at hand, and finally deploying it to production. The final output of this process is a pipeline that needs to run (and be maintained) periodically in order to keep the model up to date.

In order to cope with web-scale data sets, data scientists have resorted to *parallel and distributed computing*. MapReduce (Dean and Ghemawat, 2004) is currently the de-facto standard programming paradigm in this area, mostly thanks to the popularity of Hadoop,[1] an open source implementation of MapReduce started at Yahoo. Hadoop and its ecosystem (e.g., Mahout[2]) have proven to be an extremely successful platform to support the aforementioned process at web scale.

1. See `http://hadoop.apache.org`
2. See `http://mahout.apache.org`

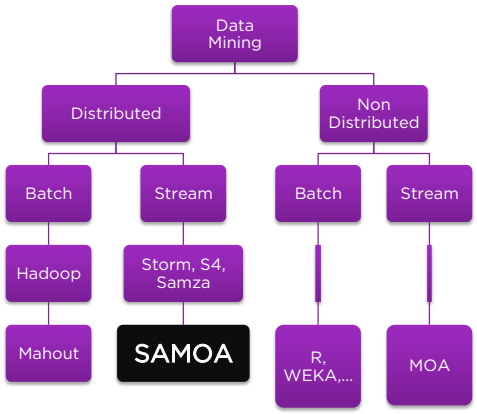


Figure 1: Taxonomy of data mining tools.

However, nowadays most data is generated in the form of a stream. Batch data is just a snapshot of streaming data obtained in an interval of time. Researchers have conceptualized and abstracted this setting in the *streaming model*. In this model data arrives at high speed, one instance at a time, and algorithms must process it in one pass under very strict constraints of space and time. Streaming algorithms make use of probabilistic guarantees to give fast approximated answers.

On the one hand, MapReduce is not suited to express streaming algorithms. On the other hand, traditional sequential online algorithms are limited by the memory and bandwidth of a single machine. *Distributed stream processing engines* (DSPEs) are a new emergent family of MapReduce-inspired technologies that address this issue. These engines allow to express parallel computation on streams, and combine the scalability of distributed processing with the efficiency of streaming algorithms. Examples of these engines include Storm,[3] S4,[4] and Samza.[5]

Alas, currently there is no common solution for mining big data streams, that is, for executing data mining and machine learning algorithms on a distributed stream processing engine. The goal of SAMOA is to fill this gap, as exemplified by Figure 1.

2. Description

SAMOA (Scalable Advanced Massive Online Analysis) is a platform for mining big data streams (De Francisci Morales, 2013). For a simple analogy, think of SAMOA as Mahout for streaming. As most of the rest of the big data ecosystem, it is written in Java.

SAMOA is both a framework and a library. As a framework, it allows algorithm developers to abstract from the underlying execution engine, and therefore reuse their code on different engines. It features a pluggable architecture that allows it to run on several distributed stream processing engines such as Storm, S4, and Samza. This capability is achieved by

3. See `http://storm.apache.org`
4. See `http://incubator.apache.org/s4`
5. See `http://samza.incubator.apache.org`

designing a minimal API that captures the essence of modern DSPEs. This API also allows to easily write new bindings to port SAMOA to new execution engines. SAMOA takes care of hiding the differences of the underlying DSPEs in terms of API and deployment.

As a library, SAMOA contains implementations of state-of-the-art algorithms for distributed machine learning on streams. For classification, SAMOA provides the Vertical Hoeffding Tree (VHT), a distributed version of a streaming decision tree (Domingos and Hulten, 2000). For clustering, it includes an algorithm based on CluStream (Aggarwal et al., 2003). For regression, a decision rule learner (Thu Vu et al., 2014). The library also includes meta-algorithms such as bagging and boosting.

The platform is intended to be useful in both research and real world deployments.

3. Architecture

An algorithm in SAMOA is represented by a directed graph of nodes that communicate via messages along streams which connect pairs of nodes. Borrowing the terminology from Storm, this graph is called a *Topology*. Each node in a Topology is a *Processor* that sends messages through a *Stream*. A Processor is a container for the code implementing the algorithm. A Stream can have a single source but several destinations (akin to a pub-sub system). A Topology is built by using a *TopologyBuilder*, which connects the various pieces of user code to the platform code and performs the necessary bookkeeping in the background. The following is a code snippet to build a topology that joins two data streams in SAMOA.

```
TopologyBuilder builder = new TopologyBuilder();
Processor sourceOne = new SourceProcessor();
builder.addProcessor(sourceOne);
Stream streamOne = builder.createStream(sourceOne);

Processor sourceTwo = new SourceProcessor();
builder.addProcessor(sourceTwo);
Stream streamTwo = builder.createStream(sourceTwo);

Processor join = new JoinProcessor();
builder.addProcessor(join).connectInputShuffle(streamOne)
    .connectInputKey(streamTwo);
```

4. Machine Learning Algorithms

The Vertical Hoeffding Tree (VHT) is a distributed extension of the VFDT (Domingos and Hulten, 2000). The VHT uses vertical parallelism to split the workload across several machines. Vertical parallelism leverages the parallelism across attributes in the same example, rather than across different examples in the stream. In practice, each training example is routed through the tree model to a leaf. There, the example is split into its constituting attributes, and each attribute is sent to a different Processor instance that keeps track of sufficient statistics. This architecture has two main advantages over one based on horizontal parallelism. First, attribute counters are not replicated across several machines, thus reducing the memory footprint. Second, the computation of the fitness of an attribute for a split decision (via, e.g., entropy or information gain) can be performed in parallel. The drawback

is that in order to get good performance, there must be sufficient inherent parallelism in the data. That is, the VHT works best for sparse data (e.g, bag-of-words models).

SAMOA includes a distributed version of CluStream, an algorithm for clustering evolving data streams. CluStream keeps a small summary of the data received so far by computing micro-clusters online. These micro-clusters are further refined to create macro-clusters by a micro-batch process, which is triggered periodically. The period can be configured via a command line parameter (e.g., every 10 000 examples).

SAMOA also includes adaptive implementations of ensemble methods such as bagging and boosting. These methods include state-of-the-art change detectors such as as `ADWIN` (Bifet and Gavaldà, 2007), DDM (Gama et al., 2004), EDDM (Baena-García et al., 2006), and Page-Hinckley (Gama et al., 2014). These meta-algorithms are most useful in conjunction with external single-machine classifiers which can be plugged in SAMOA. For instance, connectors for MOA (Bifet et al., 2010) are provided by the SAMOA-MOA package.[6]

The following listing shows how to download, build and run SAMOA.

```
# download and build SAMOA
git clone git@github.com:yahoo/samoa.git
cd samoa
mvn package

# download the Forest Cover Type data set
wget "http://downloads.sourceforge.net/project/moa-datastream/Datasets/
    Classification/covtypeNorm.arff.zip"
unzip "covtypeNorm.arff.zip"

# run SAMOA in local mode
bin/samoa local target/SAMOA-Local-0.2.0-SNAPSHOT.jar "PrequentialEvaluation
     -l classifiers.ensemble.Bagging -s (ArffFileStream -f covtypeNorm.arff)
     -f 100000"
```

5. Conclusions

SAMOA is a platform for mining big data streams. It supports the most common machine learning tasks such as classification, clustering, and regression. It also provides an API for algorithm developers that simplifies implementing distributed streaming algorithms.

SAMOA can be found at `http://www.samoa-project.net/`. The website includes a wiki, an API reference, and a developer's manual. Several examples of how the software can be used are available. The code is hosted on GitHub. SAMOA contains a test suite that is run on each commit on the GitHub repository via a continuous integration server.[7] Finally, SAMOA is released as open source software under the Apache Software License version 2.0.

We are grateful to all the people who contributed to SAMOA,[8] without whom the project could not have existed. We also thank Yahoo Labs Barcelona and its Web Mining group for the great support during the development of the project.

6. See `https://github.com/samoa-moa/samoa-moa`
7. See `https://travis-ci.org/yahoo/samoa`
8. See `http://samoa-project.net/contributors.html`

References

Charu C. Aggarwal, Jiawei Han, Jianyong Wang, and Philip S. Yu. A framework for clustering evolving data streams. In *VLDB '03: 29th International Conference on Very Large Data Bases*, pages 81–92, 2003.

Manuel Baena-García, José del Campo-Ávila, Raúl Fidalgo, Albert Bifet, Ricard Gavaldá, and Rafael Morales-Bueno. Early drift detection method. In *IWKDDS '06: 4th International Workshop on Knowledge Discovery from Data Streams*, 2006.

Albert Bifet and Ricard Gavaldà. Learning from time-changing data with adaptive windowing. In *SDM '07: Seventh SIAM International Conference on Data Mining*, pages 443–448, 2007.

Albert Bifet, Geoff Holmes, Richard Kirkby, and Bernhard Pfahringer. MOA: Massive Online Analysis. *Journal of Machine Learning Research*, 11:1601–1604, 2010.

Gianmarco De Francisci Morales. SAMOA: A platform for mining big data streams. In *RAMSS '13: 2nd International Workshop on Real-Time Analysis and Mining of Social Streams @WWW '13*, 2013.

Jeffrey Dean and Sanjay Ghemawat. Mapreduce: Simplified data processing on large clusters. In *OSDI '04: 6th Symposium on Operating Systems Design and Implementation*, pages 137–150. USENIX Association, 2004.

Pedro Domingos and Geoff Hulten. Mining high-speed data streams. In *KDD '00: 6th International Conference on Knowledge Discovery and Data Mining*, pages 71–80, 2000.

João Gama, Pedro Medas, Gladys Castillo, and Pedro Pereira Rodrigues. Learning with drift detection. In *SBIA '04: 17th Brazilian Symposium on Artificial Intelligence*, pages 286–295, 2004.

João Gama, Indre Zliobaite, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4), 2014.

Adam Jacobs. The pathologies of big data. *Communications of the ACM*, 52(8):36–44, August 2009.

Anh Thu Vu, Gianmarco De Francisci Morales, João Gama, and Albert Bifet. Distributed adaptive model rules for mining big data streams. In *BigData '14: Second IEEE International Conference on Big Data*, 2014.

Journal of Machine Learning Research 16 (2015) 155-186 Submitted 7/13; Revised 7/14; Published 2/15

Online Learning via Sequential Complexities

Alexander Rakhlin RAKHLIN@WHARTON.UPENN.EDU
Department of Statistics
University of Pennsylvania
Philadelphia, PA 19104

Karthik Sridharan SKARTHIK@WHARTON.UPENN.EDU
Department of Computer Science
Cornell University
Ithaca, NY 14853

Ambuj Tewari TEWARIA@UMICH.EDU
Department of Statistics
University of Michigan
Ann Arbor, MI 48109

Editor: Mehryar Mohri

Abstract

We consider the problem of sequential prediction and provide tools to study the minimax value of the associated game. Classical statistical learning theory provides several useful complexity measures to study learning with i.i.d. data. Our proposed sequential complexities can be seen as extensions of these measures to the sequential setting. The developed theory is shown to yield precise learning guarantees for the problem of sequential prediction. In particular, we show necessary and sufficient conditions for online learnability in the setting of supervised learning. Several examples show the utility of our framework: we can establish learnability without having to exhibit an explicit online learning algorithm.

Keywords: online learning, sequential complexities, regret minimization

1. Introduction

This paper is concerned with sequential prediction problems where no probabilistic assumptions are made regarding the data generating mechanism. Our viewpoint is expressed well by the following quotation from Cover and Shenhar (1977):

> "We are interested in sequential prediction procedures that exploit any apparent order in the sequence. We do not assume the existence of any underlying distributions, but assume that the sequence is an outcome of a game against a malevolent intelligent nature."

We will, in fact, take the game theoretic viewpoint seriously. All our investigations will proceed by analyzing the minimax value of a repeated game between a *player* or *learner* and a "malevolent intelligent nature", or the *adversary*.

Even though we have the setting of prediction problems in mind, it will be useful to develop the theory in a somewhat abstract setting. Towards this end, fix the sets $\mathcal{F}$ and

$\mathcal{Z}$, as well as a loss function $\ell : \mathcal{F} \times \mathcal{Z} \to \mathbb{R}$, and consider the following T-round repeated two-player game, which we term the *online learning* or *sequential prediction* model. On round $t \in \{1, \ldots, T\}$, the learner chooses $f_t \in \mathcal{F}$, the adversary picks $z_t \in \mathcal{Z}$, and the learner suffers loss $\ell(f_t, z_t)$. At the end of T rounds we define *regret*

$$\mathbf{R}(f_{1:T}, z_{1:T}) \triangleq \sum_{t=1}^{T} \ell(f_t, z_t) - \inf_{f \in \mathcal{F}} \sum_{t=1}^{T} \ell(f, z_t)$$

as the difference between the cumulative loss of the player and the cumulative loss of the best fixed decision. For the given pair $(\mathcal{F}, \mathcal{Z})$, the problem is said to be *online learnable* if there exists an algorithm for the learner such that regret grows sublinearly in the time horizon T, no matter what strategy the adversary employs.

The origin of the online learning (or sequential prediction) model can be traced back to the work of Robbins (1950) on compound statistical decision problems. Some of the earliest sequential prediction algorithms were proposed by Blackwell (1956a,b) and Hannan (1957). Blackwell's method was based on his celebrated approachability theorem whereas Hannan's was based on minimizing a randomly perturbed sum of previous losses. Hannan's ideas were to later resurface in the influential Follow-the-Perturbed-Leader family (Kalai and Vempala, 2005) of online learning algorithms. The seminal ideas in the work of Robbins, Blackwell and Hannan led to further developments in many different fields. Cover (1967), Davisson (1973), Ziv and Lempel (1977), Rissanen (1984), Feder et al. (1992), and others laid the foundation of universal coding, compression and prediction in the Information Theory literature. Within Computer Science, Littlestone and Warmuth (1994), Cesa-Bianchi et al. (1997), Vovk (1998), and others studied the online learning model and the prediction with expert advice framework. The connections between regret minimization and convergence to equilibria was studied in Economics by Foster and Vohra (1997), Hart and Mas-Colell (2000) and others.

We have no doubt left out many interesting works above. But even our partial list will convince the reader that research in online learning and sequential prediction has benefited from contributions by researchers from a variety of fields including Computer Science, Economics, Information Theory, and Statistics. For an excellent synthesis and presentation of results from these different fields we refer the reader to the book by Cesa-Bianchi and Lugosi (2006). Many of the ideas in the field are *constructive*, resulting in beautiful algorithms, or algorithmic techniques, associated with names such as Follow-the-Regularized-Leader, Follow-the-Perturbed-Leader, Weighted Majority, Hedge, and Online Gradient Descent. However, analyzing specific algorithms has obvious disadvantages. The algorithm may not be "optimal" for the task at hand. Even if it is optimal, one cannot prove that fact unless one develops tools for analyzing the inherent *complexity* of the online learning problem.

Our goal is precisely to provide such tools. We will begin by defining the minimax value of the game underlying the abstract online learning model. Then we will develop tools for controlling the minimax value resulting in a theory that parallels statistical learning theory. In particular, we develop analogues of combinatorial dimensions, covering numbers, and Rademacher complexities. We will also provide results relating these complexities.

Note that our approach is *non-constructive*: controlling the sequential complexities mentioned above will only guarantee the *existence* of a good online learning algorithm but

will not explicitly create one. However, it turns out that that the minimax point of view can indeed lead to constructive algorithms as shown by Rakhlin et al. (2012).

2. Minimax Value and Online Learnability

To proceed further in our analysis of the minimax value of the repeated game between the learner and the adversary, we need to make a few technical assumptions. We assume that $\mathcal{F}$ is a subset of a separable metric space. Let $\mathcal{Q}$ be the set of probability measures on $\mathcal{F}$ and assume that $\mathcal{Q}$ is weakly compact. In order to allow randomized prediction, we allow the learner to choose a distribution $q_t \in \mathcal{Q}$ on every round. The minimax value of the game is then defined as

$$\mathcal{V}_T(\mathcal{F},\mathcal{Z}) \triangleq \inf_{q_1\in\mathcal{Q}} \sup_{z_1\in\mathcal{Z}} \mathop{\mathbb{E}}_{f_1\sim q_1} \cdots \inf_{q_T\in\mathcal{Q}} \sup_{z_T\in\mathcal{Z}} \mathop{\mathbb{E}}_{f_T\sim q_T} \left[\sum_{t=1}^{T} \ell(f_t,z_t) - \inf_{f\in\mathcal{F}} \sum_{t=1}^{T} \ell(f,z_t)\right] . \tag{1}$$

Henceforth, the notation $\mathbb{E}_{f\sim q}$ stands for the expectation operator integrating out the random variable f with distribution q. We consider here the *adaptive* adversary who gets to choose each z_t based on the history of moves $f_{1:t-1}$ and $z_{1:t-1}$.

The first key step in the study of the value of the game is to appeal to the minimax theorem and exchange the pairs of infima and suprema in (1). This dual formulation is easier to analyze because the choice of the player comes *after* the choice of the mixed strategy of the adversary. We remark that the minimax theorem holds under a very general assumption of weak compactness of $\mathcal{Q}$ and lower semi-continuity of the loss function.[1] Under these conditions, we can appeal to Theorem 1 stated below, which is adapted for our needs from the work of Abernethy et al. (2009).

Theorem 1 *Let $\mathcal{F}$ and $\mathcal{Z}$ be the sets of moves for the two players, satisfying the necessary conditions for the minimax theorem to hold. Denote by $\mathcal{Q}$ and $\mathcal{P}$ the sets of probability measures (mixed strategies) on $\mathcal{F}$ and $\mathcal{Z}$, respectively. Then*

$$\mathcal{V}_T(\mathcal{F},\mathcal{Z}) = \sup_{p_1} \mathop{\mathbb{E}}_{z_1\sim p_1} \cdots \sup_{p_T} \mathop{\mathbb{E}}_{z_T\sim p_T} \left[\sum_{t=1}^{T} \inf_{f_t\in\mathcal{F}} \mathop{\mathbb{E}}_{z_t\sim p_t} [\ell(f_t,z_t)] - \inf_{f\in\mathcal{F}} \sum_{t=1}^{T} \ell(f,z_t)\right], \tag{2}$$

where suprema over p_t range over all distributions in $\mathcal{P}$.

The question of learnability in the online learning model is now reduced to the study of $\mathcal{V}_T(\mathcal{F},\mathcal{Z})$, taking (2) as the starting point.

Definition 2 *A class $\mathcal{F}$ is said to be* online learnable *with respect to the given $\mathcal{Z}$ and ℓ if*

$$\limsup_{T\to\infty} \frac{\mathcal{V}_T(\mathcal{F},\mathcal{Z})}{T} \le 0 .$$

Note that our notion of learnability is related to, but distinct from, *Hannan consistency* (Hannan, 1957; Cesa-Bianchi and Lugosi, 2006). The latter notion requires the iterated game to go on for an infinite number of rounds and is formulated in terms of *almost sure*

1. We refer to Appendix A for a precise statement of the minimax theorem, as well as sufficient conditions.

convergence. In contrast, we consider a distinct game for each T and look at *expected* regret. Nevertheless, it is possible to obtain Hannan consistency using the techniques developed in this paper by considering a slightly different game (Rakhlin et al., 2011).

We also remark that the statements in this paper extend to the case when the learner is allowed to make decisions in a larger set $\mathcal{G}$, while the best-in-hindsight term in the regret definition is computed with respect to $\mathcal{F} \subseteq \mathcal{G}$. Such a setting—interesting especially with regard to computational concerns—is termed *improper learning*. For example, prediction with side information (or, the *supervised learning* problem) is one such case, where we choose $\mathcal{Y} \subset \mathbb{R}$, $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}} = \mathcal{G}$ and $\ell(f,(x,y)) = |f(x) - y|$. This setting will be studied later in the paper. Note that in the proper learning scenario, $\mathcal{V}_T(\mathcal{F},\mathcal{Z}) \geq 0$ (e.g., since all z_t's can be chosen to be the same), and thus the "lim sup" in Definition 2 can be simply replaced with the limit being equal to zero.

This paper is aimed at understanding the value of the game $\mathcal{V}_T(\mathcal{F},\mathcal{Z})$ for various function classes $\mathcal{F}$. Since our focus is on the complexity of $\mathcal{F}$, we shall often write $\mathcal{V}_T(\mathcal{F})$ keeping the dependence on $\mathcal{Z}$ (and ℓ) implicit. As we show, the sequential complexity notions—that were shown by Rakhlin et al. (2014) to characterize uniform martingale Laws of Large Numbers—also give us a handle on the value $\mathcal{V}_T(\mathcal{F})$. In the next section, we briefly define these sequential complexity notions and mention some of the key relations between them. A more detailed account of the relationships between sequential complexity measures along with complete proofs can be found in (Rakhlin et al., 2014).

3. Sequential Complexities

Unlike the well-studied statistical learning scenario with i.i.d. data, the online learning problem possesses a certain sequential dependence. Such dependence cannot be captured by classical notions of complexity that are based on a batch of data given as a *tuple* of T examples. A basic unit that does capture temporal dependence is a binary tree. Surprisingly, for the sequential prediction problems considered in this paper, one need not look further than binary trees to capture the relevant complexity.

A *$\mathcal{Z}$-valued tree* $\mathbf{z}$ *of depth* T is a complete rooted binary tree with nodes labeled by elements of $\mathcal{Z}$. Such a tree $\mathbf{z}$ is identified with the sequence $(\mathbf{z}_1,\dots,\mathbf{z}_T)$ of labeling functions $\mathbf{z}_i : \{\pm 1\}^{i-1} \to \mathcal{Z}$ which provide the labels for each node. Therefore, $\mathbf{z}_1 \in \mathcal{Z}$ is the label for the *root* of the tree, while $\mathbf{z}_i$ for $i > 1$ is the label of the node obtained by following the path of length $i-1$ from the root, with $+1$ indicating 'right' and -1 indicating 'left'. A *path* of length T is given by the sequence $\epsilon = (\epsilon_1,\dots,\epsilon_T) \in \{\pm 1\}^T$. For brevity, we shall often write $\mathbf{z}_t(\epsilon)$, where $\epsilon = (\epsilon_1,\dots,\epsilon_T)$, but it is understood that $\mathbf{z}_t$ depends only on the prefix $(\epsilon_1,\dots,\epsilon_{t-1})$.

Now, let $\epsilon_1,\dots,\epsilon_T$ be independent Rademacher random variables. Given a $\mathcal{Z}$-valued tree $\mathbf{z}$ of depth T, we define the *sequential Rademacher complexity* of a function class $\mathcal{G} \subseteq \mathbb{R}^{\mathcal{Z}}$ on a $\mathcal{Z}$-valued tree $\mathbf{z}$ as

$$\mathfrak{R}_T(\mathcal{G},\mathbf{z}) \triangleq \mathbb{E}\left[\sup_{g\in\mathcal{G}} \frac{1}{T}\sum_{t=1}^{T} \epsilon_t g(\mathbf{z}_t(\epsilon))\right],$$

and we denote by $\mathfrak{R}_T(\mathcal{G}) = \sup_{\mathbf{z}} \mathfrak{R}_T(\mathcal{G},\mathbf{z})$ its supremum over all $\mathcal{Z}$-valued trees of depth T. The importance of the introduced notion stems from the following result (Rakhlin et al.,

2014, Theorem 2): for any distribution over a sequence $(Z_1, \ldots, Z_T)$, we have

$$\mathbb{E}\left[\sup_{g\in\mathcal{G}} \frac{1}{T}\sum_{t=1}^{T}\left(\mathbb{E}\left[g(Z_t)|Z^{t-1}\right] - g(Z_t)\right)\right] \le 2\,\mathfrak{R}_T(\mathcal{G}) \ , \tag{3}$$

where $Z^{t-1} = (Z_1, \ldots, Z_{t-1})$. In other words, the martingale version of the uniform deviations of means from expectations is controlled by the worst-case sequential Rademacher complexity. A matching lower bound also holds for the supremum over distributions on sequences in $\mathcal{Z}^T$. It then follows that a uniform martingale Law of Large Numbers holds for $\mathcal{G}$ if and only if $\mathfrak{R}_T(\mathcal{G}) \to 0$. For i.i.d. random variables, a similar statement can be made in terms of the classical Rademacher complexity, and so one might hope that many other complexity notions from empirical process theory have martingale (or we may say, sequential) analogues. Luckily, this is indeed the case (Rakhlin et al., 2014). As we show in this paper, these generalizations of the classical notions also give a handle on (as well as necessary and sufficient conditions for) online learnability, thus painting a picture that completely parallels statistical learning theory. But before we present our main results, let us recall some key definitions and results in (Rakhlin et al., 2014).

In providing further upper bounds on sequential Rademacher complexity, the following definitions of an "effective size" of a function class generalize the classical notions of a covering number. A set V of $\mathbb{R}$-valued trees of depth T is *a (sequential) α-cover* (with respect to ℓ_p norm) of $\mathcal{G} \subseteq \mathbb{R}^{\mathcal{Z}}$ on a tree $\mathbf{z}$ of depth T if

$$\forall g \in \mathcal{G},\ \forall \epsilon \in \{\pm 1\}^T,\ \exists \mathbf{v} \in V \text{ s.t.} \quad \left(\frac{1}{T}\sum_{t=1}^{T} |\mathbf{v}_t(\epsilon) - g(\mathbf{z}_t(\epsilon))|^p\right)^{1/p} \le \alpha.$$

The *(sequential) covering number* of a function class $\mathcal{G}$ on a given tree $\mathbf{z}$ is defined as

$$\mathcal{N}_p(\alpha, \mathcal{G}, \mathbf{z}) \ \triangleq\ \min\left\{|V| : V \text{ is an } \alpha\text{-cover w.r.t. } \ell_p \text{ norm of } \mathcal{G} \text{ on } \mathbf{z}\right\}.$$

It is straightforward to check that $\mathcal{N}_p(\alpha, \mathcal{G}, \mathbf{z}) \le \mathcal{N}_q(\alpha, \mathcal{G}, \mathbf{z})$ whenever $1 \le p \le q \le \infty$.

Further define $\mathcal{N}_p(\alpha, \mathcal{G}, T) = \sup_{\mathbf{z}} \mathcal{N}_p(\alpha, \mathcal{G}, \mathbf{z})$, the maximal ℓ_p covering number of $\mathcal{G}$ over depth T trees. For a class $\mathcal{G}$ of binary-valued functions, we also define a so-called 0-*cover* (or, cover at scale 0), denoted by $\mathcal{N}(0, \mathcal{G}, \mathbf{z})$, as equal to any $\mathcal{N}_p(0, \mathcal{G}, \mathbf{z})$. The definition of a 0-cover can be seen as the correct analogue of the *size of a projection* of $\mathcal{G}$ onto a tuple of points in the i.i.d. case. The size of this projection in the i.i.d. case was the starting point of the work of Vapnik and Chervonenkis.

When $\mathcal{G} \subseteq [-1, 1]^{\mathcal{Z}}$ is a *finite* class of bounded functions, one can show (Rakhlin et al., 2014, Lemma 1) that

$$\mathfrak{R}_T(\mathcal{G}, \mathbf{z}) \le \sqrt{\frac{2\log|\mathcal{G}|}{T}}, \tag{4}$$

a bound that should (correctly) remind the reader of the Exponential Weights regret bound. With the definition of an α-cover with respect to ℓ_1 norm, one can easily extend (4) beyond the finite case. Immediately from the definition of ℓ_1 covering number, it follows that for any $\mathcal{G} \subseteq [-1, 1]^{\mathcal{Z}}$, for any $\alpha > 0$,

$$\mathfrak{R}_T(\mathcal{G}, \mathbf{z}) \le \alpha + \sqrt{\frac{2\log\mathcal{N}_1(\alpha, \mathcal{G}, \mathbf{z})}{T}} \tag{5}$$

(Rakhlin et al., 2014, Eq. 9). A tighter control is obtained by integrating the covering numbers at different scales. To this end, consider the following analogue of the Dudley entropy integral bound. For $p \geq 1$, the *integrated complexity* of a function class $\mathcal{G} \subseteq [-1,1]^{\mathcal{Z}}$ on a $\mathcal{Z}$-valued tree of depth T is defined as

$$\mathfrak{D}_T^p(\mathcal{G}, \mathbf{z}) \triangleq \inf_{\alpha \geq 0} \left\{ 4\alpha + \frac{12}{\sqrt{T}} \int_\alpha^1 \sqrt{\log \mathcal{N}_p(\delta, \mathcal{G}, \mathbf{z})}\, d\delta \right\} \tag{6}$$

and $\mathfrak{D}_T^p(\mathcal{G}) = \sup_{\mathbf{z}} \mathfrak{D}_T^p(\mathcal{G}, \mathbf{z})$, with $\mathfrak{D}_T^2(\mathcal{G}, \mathbf{z})$ denoted simply by $\mathfrak{D}_T(\mathcal{G}, \mathbf{z})$. We have previously shown (Rakhlin et al., 2014, Theorem 3) that, for any function class $\mathcal{G} \subseteq [-1,1]^{\mathcal{Z}}$ and any $\mathcal{Z}$-valued tree $\mathbf{z}$ of depth T,

$$\mathfrak{R}_T(\mathcal{G}, \mathbf{z}) \leq \mathfrak{D}_T(\mathcal{G}, \mathbf{z}). \tag{7}$$

We next turn to the description of sequential combinatorial parameters. A $\mathcal{Z}$-valued tree $\mathbf{z}$ of depth d is *shattered* by a function class $\mathcal{G} \subseteq \{\pm 1\}^{\mathcal{Z}}$ if for all $\epsilon \in \{\pm 1\}^d$, there exists $g \in \mathcal{G}$ such that $g(\mathbf{z}_t(\epsilon)) = \epsilon_t$ for all $t \in [d]$. The *Littlestone dimension* $\mathrm{Ldim}(\mathcal{G}, \mathcal{Z})$ is the largest positive integer d such that $\mathcal{G}$ shatters a $\mathcal{Z}$-valued tree of depth d (Littlestone, 1988; Ben-David et al., 2009). The scale-sensitive version of Littlestone dimension is defined as follows. A $\mathcal{Z}$-valued tree $\mathbf{z}$ of depth d is *α-shattered* by a function class $\mathcal{G} \subseteq \mathbb{R}^{\mathcal{Z}}$ if there exists an $\mathbb{R}$-valued tree $\mathbf{s}$ of depth d such that

$$\forall \epsilon \in \{\pm 1\}^d,\ \exists g \in \mathcal{G} \quad \text{s.t. } \forall t \in [d],\ \epsilon_t (g(\mathbf{z}_t(\epsilon)) - \mathbf{s}_t(\epsilon)) \geq \alpha/2.$$

The tree $\mathbf{s}$ will be called a *witness to shattering*. The *(sequential) fat-shattering dimension* $\mathrm{fat}_\alpha(\mathcal{G}, \mathcal{Z})$ at scale α is the largest d such that $\mathcal{G}$ α-shatters a $\mathcal{Z}$-valued tree of depth d.

The notions introduced above can be viewed as sequential generalizations of the VC dimension and the fat-shattering dimension where tuples of points get replaced by complete binary trees. In fact, one recovers the classical notions if the tree $\mathbf{z}$ in the above definitions is restricted to have the same values within a level (hence, no temporal dependence). Crucially, the sequential combinatorial analogues provide control for the growth of sequential covering numbers, justifying the definitions.

First, let $\mathcal{G} \subseteq \{0, \dots, k\}^{\mathcal{Z}}$ be a class of functions with $\mathrm{fat}_2(\mathcal{G}) = d$. Then, it can be shown (Rakhlin et al., 2014, Theorem 4) that for any $T \geq 1$,

$$\mathcal{N}_\infty(1/2, \mathcal{G}, T) \leq \sum_{i=0}^{d} \binom{T}{i} k^i \leq (ekT)^d .$$

For the second result (Rakhlin et al., 2014, Corollary 1), suppose $\mathcal{G}$ is a class of $[-1,1]$-valued functions on $\mathcal{Z}$. Then, for any $\alpha > 0$, and any $T \geq 1$,

$$\mathcal{N}_\infty(\alpha, \mathcal{G}, T) \leq \left(\frac{2eT}{\alpha}\right)^{\mathrm{fat}_\alpha(\mathcal{G})} . \tag{8}$$

Finally, we recall a bound on the size of the 0-cover in terms of the fat_1 combinatorial parameter (Rakhlin et al., 2014, Theorem 5). For a class $\mathcal{G} \subseteq \{0, \dots, k\}^{\mathcal{Z}}$ with $\mathrm{fat}_1(\mathcal{G}) = d$, we have

$$\mathcal{N}(0, \mathcal{G}, T) \leq \sum_{i=0}^{d} \binom{T}{i} k^i \leq (ekT)^d \ . \tag{9}$$

In particular, for $k = 1$ (that is, binary classification) we have $\mathrm{fat}_1(\mathcal{G}) = \mathrm{Ldim}(\mathcal{G})$. The inequality (9) is therefore a sequential analogue of the celebrated Vapnik-Chervonenkis-Sauer-Shelah lemma.

4. Structural Properties

For the examples developed in this paper, it will be crucial to exploit a number of useful properties that $\mathfrak{R}_T(\mathcal{G})$ satisfies. These properties allow one to establish online learnability for complex function classes even if no explicit learning algorithms are available.

We first state some properties that are easily proved but are nevertheless very useful.

Lemma 3 *Let $\mathcal{F}, \mathcal{G} \subseteq \mathbb{R}^{\mathcal{Z}}$ and let $\mathrm{conv}(\mathcal{G})$ denote the convex hull of $\mathcal{G}$. Let $\mathbf{z}$ be any $\mathcal{Z}$-valued tree of depth T. Then the following properties hold.*

1. *If $\mathcal{F} \subseteq \mathcal{G}$, then $\mathfrak{R}_T(\mathcal{F}, \mathbf{z}) \le \mathfrak{R}_T(\mathcal{G}, \mathbf{z})$.*
2. *$\mathfrak{R}_T(\mathrm{conv}(\mathcal{G}), \mathbf{z}) = \mathfrak{R}_T(\mathcal{G}, \mathbf{z})$*
3. *$\mathfrak{R}_T(c\mathcal{G}, \mathbf{z}) = |c|\mathfrak{R}_T(\mathcal{G}, \mathbf{z})$ for all $c \in \mathbb{R}$.*
4. *For any $h : \mathcal{Z} \to \mathbb{R}$, $\mathfrak{R}_T(\mathcal{G} + h, \mathbf{z}) = \mathfrak{R}_T(\mathcal{G}, \mathbf{z})$ where $\mathcal{G} + h = \{g + h : g \in \mathcal{G}\}$.*

These properties match those of the classical Rademacher complexity (Bartlett and Mendelson, 2003) and can be proved in essentially the same way (we therefore skip the straightforward proofs).

The next property is a key tool for many of the applications: it allows us to bound the sequential Rademacher complexity for the Cartesian product of function classes composed with a Lipschitz mapping in terms of complexities of the individual classes.

Lemma 4 *Let $\mathcal{G} = \mathcal{G}_1 \times \ldots \times \mathcal{G}_k$ where each $\mathcal{G}_j \subseteq [-1,1]^{\mathcal{Z}}$. Further, let $\phi : \mathbb{R}^k \times \mathcal{Z} \to \mathbb{R}$ be such that $\phi(\cdot, z)$ is L-Lipschitz with respect to $\|\cdot\|_\infty$ for all $z \in \mathcal{Z}$, and let*

$$\phi \circ \mathcal{G} = \{z \mapsto \phi((g_1(z), \ldots, g_k(z)), z) : g_j \in \mathcal{G}_j\}.$$

Then we have

$$\mathfrak{R}_T(\phi \circ \mathcal{G}) \le 8\, L \left(1 + 4\sqrt{2}\log^{3/2}(eT^2)\right) \textstyle\sum_{j=1}^k \mathfrak{R}_T(\mathcal{G}_j)$$

as long as $\mathfrak{R}_T(\mathcal{G}_j) \ge 1/T$ for each j.

Let us explicitly state the more familiar contraction property, an immediate corollary of the above result.

Corollary 5 *Fix a class $\mathcal{G} \subseteq [-1,1]^{\mathcal{Z}}$ with $\mathfrak{R}_T(\mathcal{G}) \ge 1/T$ and a function $\phi : \mathbb{R} \times \mathcal{Z} \to \mathbb{R}$. Assume $\phi(\cdot, z)$ is L-Lipschitz for all $z \in \mathcal{Z}$. Then*

$$\mathfrak{R}_T(\phi \circ \mathcal{G}) \le 8\, L \left(1 + 4\sqrt{2}\log^{3/2}(eT^2)\right) \cdot \mathfrak{R}_T(\mathcal{G}),$$

where $\phi \circ \mathcal{G} = \{z \mapsto \phi(g(z), z) : g \in \mathcal{G}\}$.

We state another useful corollary of Lemma 4.

Corollary 6 *For a fixed binary function $b : \{\pm 1\}^k \to \{\pm 1\}$ and classes $\mathcal{G}_1, \dots, \mathcal{G}_k$ of $\{\pm 1\}$-valued functions,*

$$\mathfrak{R}_T(b(\mathcal{G}_1, \dots, \mathcal{G}_k)) \le \mathcal{O}\left(\log^{3/2}(T)\right) \sum_{j=1}^k \mathfrak{R}_T(\mathcal{G}_j).$$

Note that, in the classical case, the Lipschitz contraction property holds without any extra poly-logarithmic factors in T (Ledoux and Talagrand, 1991). It is an open question whether the poly-logarithmic factors can be removed in the results above. It is worth pointing out ahead of time that Theorem 8 below—in the setting of supervised learning with convex Lipschitz loss—does allow us to avoid the extraneous factor that would otherwise appear from a combination of Theorem 7 and Corollary 5.

5. Main Results

We now relate the value of the game to the worst case expected value of the supremum of an empirical process. However, unlike empirical processes that involve i.i.d. sums, our process involves a sum of *martingale differences*. In view of (3), the expected supremum can be further upper-bounded by the sequential Rademacher complexity.

Theorem 7 *The minimax value is bounded as*

$$\frac{1}{T}\mathcal{V}_T(\mathcal{F}) \le \sup_{\mathbb{P}} \mathbb{E} \sup_{g \in \ell(\mathcal{F})} \left[\frac{1}{T} \sum_{t=1}^T \left(\mathbb{E}[g(Z_t)|Z_1, \dots, Z_{t-1}] - g(Z_t) \right) \right] \le 2\,\mathfrak{R}_T(\ell(\mathcal{F})),$$

where $\ell(\mathcal{F}) = \{\ell(f, \cdot) : f \in \mathcal{F}\}$ and the supremum is taken over all distributions $\mathbb{P}$ over $(Z_1, \dots, Z_T)$.

We can now employ the tools developed earlier in the paper to upper bound the value of the game. Interestingly, any non-trivial upper bound guarantees *existence* of a prediction strategy that has sublinear regret irrespective of the sequence of the moves of the adversary. This complexity-based approach of establishing learnability should be contrasted with the purely algorithm-based approaches found in the literature.

5.1 Supervised Learning

In this subsection we study the *supervised learning problem* mentioned earlier in the paper. In this improper learning scenario, the learner at time t picks a function $f_t : \mathcal{X} \to \mathbb{R}$ and the adversary provides the input target pair $z_t = (x_t, y_t) \in \mathcal{X} \times \mathcal{Y}$ where $\mathcal{Y} \subset \mathbb{R}$. In particular, the *binary classification* problem corresponds to the case $\mathcal{Y} = \{\pm 1\}$. Let $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$ and let us fix the absolute value loss function $\ell(\hat{y}, y) = |\hat{y} - y|$. While we focus on the absolute loss, it is easy to see that all the results hold (with modified rates) for any loss $\ell(\hat{y}, y)$ such that for all $\hat{y}$ and y, $\phi(\ell(\hat{y}, y)) \le |\hat{y} - y| \le \Phi(\ell(\hat{y}, y))$ where Φ and ϕ are monotonically increasing functions. For instance, the squared loss $(\hat{y} - y)^2$ is a classic example.

We now observe that the value of the improper supervised learning game can be equivalently written as

$$\mathcal{V}_T^{\mathrm{S}}(\mathcal{F}) = \sup_{x_1} \inf_{q_1 \in \tilde{\mathcal{Q}}} \sup_{y_1} \mathop{\mathbb{E}}_{\hat{y}_1 \sim q_1} \cdots \sup_{x_T} \inf_{q_T \in \tilde{\mathcal{Q}}} \sup_{y_T} \mathop{\mathbb{E}}_{\hat{y}_T \sim q_T} \left[\sum_{t=1}^T \ell(\hat{y}_t, y_t) - \inf_{f \in \mathcal{F}} \sum_{t=1}^T \ell(f(x_t), y_t) \right], \qquad (10)$$

where $\tilde{\mathcal{Q}}$ denotes the set of probability distributions over $\mathcal{Y}$ and $\hat{y}_t$ has distribution q_t. This equivalence is easy to verify: we may view the choice $f_t : \mathcal{X} \to \mathcal{Y}$ as pre-specifying predictions $f_t(x)$ for all the possible $x \in \mathcal{X}$, while alternatively we can simply make the choice $\hat{y}_t \in \mathcal{Y}$ having observed the particular move $x_t \in \mathcal{X}$. The advantage of rewriting the game in the form (10) is that the minimax theorem only needs to be applied to the pair $\hat{y}_t$ and y_t, given the fixed choice x_t. The minimax theorem then holds even if weak compactness cannot be shown for the set of distributions on the original space of functions of the type $\mathcal{X} \to \mathcal{Y}$.

An examination of the proof of Theorem 7 reveals that the value (10) is upper bounded in exactly the same way, and the side information simply appears as an additional tree $\mathbf{x}$ in sequential Rademacher complexity, giving us:

$$\frac{1}{T}\mathcal{V}_T^{\mathrm{S}}(\mathcal{F}) \le 2 \sup_{\mathbf{x},\mathbf{y}} \mathbb{E}\left[\sup_{f\in\mathcal{F}} \frac{1}{T}\sum_{t=1}^{T} \epsilon_t \ell(f(\mathbf{x}_t(\epsilon)), \mathbf{y}_t(\epsilon))\right] . \tag{11}$$

However, for the supervised learning setting, we can strengthen Theorem 7. The following theorem allows us to remove any convex Lipschitz loss (including the absolute loss) before passing to the sequential Rademacher complexity.

Theorem 8 *Let $\mathcal{Y} = [-1, 1]$ and suppose, for any $y \in \mathcal{Y}$, $\ell(\cdot, y)$ is convex and L-Lipschitz. Then the minimax value of a supervised learning problem is upper bounded as*

$$\frac{1}{T}\mathcal{V}_T^{S}(\mathcal{F}) \le 2\, L\, \mathfrak{R}_T(\mathcal{F}).$$

We remark that the contraction property for sequential Rademacher complexity, stated in Section 4, yields an extraneous logarithmic factor when applied to (11); here, we achieve the desired bound by removing the Lipschitz function directly during the symmetrization step.

Armed with the theorem, we now prove the following result.

Proposition 9 *Consider the supervised learning problem with a function class $\mathcal{F} \subseteq [-1, 1]^{\mathcal{X}}$ and absolute loss $\ell(\hat{y}, y) = |\hat{y} - y|$. Then, for any $T \ge 1$, we have*

$$\begin{aligned}\frac{1}{4\sqrt{2}} \sup_{\alpha}\left\{\alpha\sqrt{\frac{\min\{\mathrm{fat}_\alpha, T\}}{T}}\right\} \le \mathfrak{R}_T(\mathcal{F}) \le \frac{1}{T}\mathcal{V}_T^{S}(\mathcal{F}) \le 2\mathfrak{R}_T(\mathcal{F}) \le 2\mathfrak{D}_T(\mathcal{F}) \\ \le\ 2\inf_{\alpha}\left\{4\alpha + \frac{12}{\sqrt{T}}\int_{\alpha}^{1}\sqrt{\mathrm{fat}_\beta \log\left(\frac{2eT}{\beta}\right)}\, d\beta\right\} ,\end{aligned} \tag{12}$$

where $\mathrm{fat}_\alpha = \mathrm{fat}_\alpha(\mathcal{F})$.

The proposition above implies that finiteness of the fat-shattering dimension at all scales is *necessary and sufficient* for online learnability of the supervised learning problem. Further, all the complexity notions introduced so far are within a poly-logarithmic factor from each other whenever the problem is learnable. These results are summarized in the next theorem:

Theorem 10 *For any function class $\mathcal{F} \subseteq [-1, 1]^{\mathcal{X}}$, the following statements are equivalent*

1. *Function class $\mathcal{F}$ is online learnable in the supervised setting with absolute loss.*
2. *Sequential Rademacher complexity satisfies* $\lim_{T\to\infty} \mathfrak{R}_T(\mathcal{F}) = 0$.
3. *For any* $\alpha > 0$, *the scale-sensitive dimension* $\mathrm{fat}_\alpha(\mathcal{F})$ *is finite.*

Moreover, if the function class is online learnable, then the value of the supervised game $\mathcal{V}_T^S(\mathcal{F})$, *the sequential Rademacher complexity* $\mathfrak{R}_T(\mathcal{F})$, *and the integrated complexity* $\mathfrak{D}_T(\mathcal{F})$ *are within a multiplicative factor of* $\mathcal{O}(\log^{3/2} T)$ *of each other.*

Remark 11 *Additionally, the three statements of Theorem 10 are equivalent to $\mathcal{F}$ satisfying a martingale version of the uniform Law of Large Numbers. This property is termed* Sequential Uniform Convergence *by Rakhlin et al. (2014), and we refer to their paper for more details.*

For binary classification, we write $\mathcal{V}_T^{\mathrm{Binary}}$ for $\mathcal{V}_T^{\mathrm{S}}$. This case has been investigated thoroughly by Ben-David et al. (2009) and indeed served as a key motivation for this paper. As a consequence of Proposition 9 and (9), we have a tight control on the value of the game for the binary classification problem. Note that the absolute loss in the binary classification setting is simply the 0-1 loss $\ell(\hat{y}, y) = \mathbf{1}\{\hat{y} \neq y\}$, where $\mathbf{1}\{\mathcal{U}\}$ is 1 if $\mathcal{U}$ is true and 0 otherwise.

Corollary 12 *For the binary classification problem with function class $\mathcal{F}$ and the* 0-1 *loss, we have*

$$K_1\sqrt{T \min\{\mathrm{Ldim}(\mathcal{F}), T\}} \le \mathcal{V}_T^{\mathrm{Binary}}(\mathcal{F}) \le K_2\sqrt{T\ \mathrm{Ldim}(\mathcal{F}) \log T}$$

for some universal constants $K_1, K_2 > 0$.

Both the upper and the lower bound in the above result were originally derived in Ben-David et al. (2009). Notably, we achieved the same bounds non-constructively through purely combinatorial and covering number arguments.

It is natural to ask whether being able to learn in the online model is different from learning in the i.i.d. model (in the distribution-free supervised setting). The standard example that exhibits a gap between the two frameworks (e.g., Littlestone, 1988; Ben-David et al., 2009) is binary classification using the class of step functions

$$\mathcal{F} = \{f_\theta(x) = \mathbf{1}\{x \le \theta\} : \theta \in [0,1]\}$$

on $[0,1]$. This class has VC dimension 1, but is *not* learnable in the online setting. Indeed, it is possible to verify that the Littlestone dimension is infinite. Interestingly, the closely-related class of "ramp" functions with slope $L > 0$

$$\mathcal{F}_L = \left\{f_\theta(x) = \mathbf{1}\{x \le \theta\} + (1 - L(x-\theta))\mathbf{1}\{\theta < x \le \theta + 1/L\} : \theta \in [0,1]\right\}$$

is learnable (say for supervised learning using absolute loss) in the online setting (and hence also in the i.i.d. case). Furthermore, the larger class of all bounded *L-Lipschitz* functions on a bounded interval is also online learnable (see Eq. 14 and proof of Proposition 18). Once again, we are able to make these statements from purely complexity-based considerations, without exhibiting an algorithm. Further examples where we can demonstrate online learnability are explored in Section 6.

5.2 Online Convex Optimization

Over the past decade, Online Convex Optimization (OCO) has emerged as a unified online learning framework (Zinkevich, 2003; Shalev-Shwartz, 2011). Various methods, such as Exponential Weights, can be viewed as instances of online mirror descent, solving the associated OCO problem. Much research effort has been devoted to understanding this abstract and simplified setting. It is tempting to say that any problem of online learning, as defined in the Introduction, can be viewed as OCO (in fact, online *linear* optimization) over the set of probability distributions; however, one should also recognize that by linearizing the problem, any interesting structure is lost and one instead suffers from the possibly unnecessary dependence on the number of functions in the class $\mathcal{F}$. Nevertheless, OCO is a central part of the recent literature, and we will study this scenario using techniques developed in this paper.

The standard setting of online convex optimization is as follows. The set of moves of the learner $\mathcal{F}$ is a bounded closed convex subset of a Banach space $(\mathcal{B}, \|\cdot\|)$ with $\|f\| \le D$ for all $f \in \mathcal{F}$ (the reader can think of $\mathbb{R}^d$ equipped with an ℓ_p norm for simplicity). Let $\|\cdot\|_\star$ be the dual norm. The adversary's set $\mathcal{Z}$ consists of convex G-Lipschitz (with respect to $\|\cdot\|_\star$) functions over $\mathcal{F}$:

$$\mathcal{Z} = \mathcal{Z}_{\mathrm{cvx}} = \{g : \mathcal{F} \to \mathbb{R} \,:\, g \text{ convex and } G\text{-Lipschitz w.r.t. } \|\cdot\|_\star\} \ .$$

Let the loss function be $\ell(f, g) = g(f)$, the evaluation of the adversarially chosen function at f. For the particular case of online *linear* optimization, we instead take

$$\mathcal{Z} = \mathcal{Z}_{\mathrm{lin}} = \{f \mapsto \langle f, z\rangle \,:\, \|z\|_\star \le G\}$$

with $\mathcal{Z}$ now a subset of the dual space. It is well-known (e.g., Abernethy et al., 2008) that the online convex optimization problem (without further assumptions on the functions in $\mathcal{Z}_{\mathrm{cvx}}$) is as hard as the corresponding linear optimization problem with $\mathcal{Z}_{\mathrm{lin}}$ if one considers deterministic algorithms. The same trivially extends to randomized methods:

Lemma 13 *Suppose $\mathcal{F}, \mathcal{Z}_{\mathrm{cvx}}, \mathcal{Z}_{\mathrm{lin}}$ be defined as above. Then we have*

$$\mathcal{V}_T(\mathcal{F}, \mathcal{Z}_{\mathrm{cvx}}) = \mathcal{V}_T(\mathcal{F}, \mathcal{Z}_{\mathrm{lin}}) \ .$$

We will now show how to use the above result to derive minimax regret guarantees for OCO. The reader may wonder why we do not directly try to bound the value $\mathcal{V}_T(\mathcal{F}, \mathcal{Z}_{\mathrm{cvx}})$ by $\mathfrak{R}_T(\mathcal{F}, \mathcal{Z}_{\mathrm{cvx}})$. In fact, this proof strategy cannot give a non-trivial bound if $\mathcal{F}$ is a subset of a high-dimensional (or infinite-dimensional) space (Shalev-Shwartz et al., 2009, Sec. 4.1). Instead, we use the lemma above to bound the value of the game where adversary plays convex functions with that of the game where adversary plays linear functions.

A function $\Psi : \mathcal{F} \to \mathbb{R}$ is (σ, q)-uniformly convex (for $q \in [2, \infty)$) on $\mathcal{F}$ with respect to a norm $\|\cdot\|$ if, for all $\theta \in [0, 1]$ and $f_1, f_2 \in \mathcal{F}$,

$$\Psi(\theta f_1 + (1-\theta) f_2) \le \theta \Psi(f_1) + (1-\theta)\Psi(f_2) - \frac{\sigma\, \theta\, (1-\theta)}{q} \|f_1 - f_2\|^q \ .$$

A $(\sigma, 2)$-uniformly convex function will be called σ-strongly convex.

We will give examples shortly but we first state a proposition that is useful to bound the sequential Rademacher complexity of linear function classes. The crucial duality fact exploited in its proof is that Ψ is (σ, q)-uniformly convex with respect to $\|\cdot\|$ if and only if $\Psi^\star$ is $(1/\sigma, p)$-uniformly smooth with respect to $\|\cdot\|_\star$ where $1/p + 1/q = 1$.

Proposition 14 (Rakhlin et al., 2014) *Let $\mathcal{F}$ be a subset of some Banach space $\mathcal{B}$ with norm $\|\cdot\|$ and let $\mathcal{Z}$ be a subset of the dual space $\mathcal{B}^\star$ equipped with norm $\|\cdot\|_\star$. Suppose that $\Psi : \mathcal{F} \to \mathbb{R}$ is (σ, q)-uniformly convex with respect to $\|\cdot\|$ and $0 \le \Psi(f) \le \Psi_{\max}$ for all $f \in \mathcal{F}$. Then we have*

$$\mathfrak{R}_T(\mathcal{F}) \le C_p \|\mathcal{Z}\|_\star \left(\frac{\Psi_{\max}^{p-1}}{\sigma T^{p-1}} \right)^{1/p},$$

where $\|\mathcal{Z}\|_\star = \sup_{z \in \mathcal{Z}} \|z\|_\star$, p is such that $1/p + 1/q = 1$, and $C_p = (p/(p-1))^{\frac{p-1}{p}}$.

Using the above Proposition in conjunction with Lemma 13 and Theorem 7, we can immediately conclude that

$$\mathcal{V}_T(\mathcal{F}, \mathcal{Z}_{\text{cvx}}) \le 2\,T\,\mathfrak{R}_T(\mathcal{F}) \le 2G\sqrt{\frac{2\,\Psi_{\max}\,T}{\sigma}}$$

for any non-negative function $\Psi : \mathcal{F} \to \mathbb{R}$ that is σ-strongly convex w.r.t. $\|\cdot\|$. Note that, typically, $\Psi_{\max}$ will depend on D. For example, in the particular case when $\|\cdot\| = \|\cdot\|_\star = \|\cdot\|_2$, we can take $\Psi(u) = \frac{1}{2}\|u\|_2^2$ and the above bound becomes $2GD\sqrt{T}$ and recovers the guarantee for the online gradient descent algorithm. In general, for $\|\cdot\| = \|\cdot\|_p$ and $\|\cdot\|_\star = \|\cdot\|_q$, we can use $\Psi(u) = \frac{1}{2}\|u\|_p^2$ to get a bound of $2GD\sqrt{T/(p-1)}$ since Ψ is $(p-1)$-strongly convex w.r.t. $\|\cdot\|_p$. These $\mathcal{O}(\sqrt{T})$ regret rates are not new but we re-derive them to illustrate the usefulness of the tools we developed.

6. Further Examples

Now we present some further applications of the tools we have developed in this paper for some specific learning problems. To begin, we show how to bound the sequential Rademacher complexity of functions computed by neural networks. Then, we derive margin based regret bounds in a fairly general setting. The classical analogues of these margin bounds have played a big role in the modern theory of supervised learning where they help explain the success of linear classifiers in high dimensional spaces (e.g., Schapire et al., 1997; Koltchinskii and Panchenko, 2002). We then study the complexity of classes formed by decision trees, analyze the setting of transductive learning, and consider an online version of the Isotron problem. Finally, we make a connection to the seminal work of Cesa-Bianchi and Lugosi (1999) by re-deriving their bound on the minimax regret in a static experts game in terms of the classical Rademacher averages.

6.1 Neural Networks

We provide below a bound on the sequential Rademacher complexity for classic multi-layer neural networks thus showing they are learnable in the online setting. The model of neural

networks we consider below and the bounds we provide are analogous to the ones considered in the i.i.d. setting by Bartlett and Mendelson (2003).

Consider a k-layer 1-norm neural network, defined by a base function class $\mathcal{F}_1$ and, recursively, for each $2 \le i \le k$,

$$\mathcal{F}_i = \left\{ x \mapsto \sum_j w_j^i \sigma\left(f_j(x)\right) \;\middle|\; \forall j \; f_j \in \mathcal{F}_{i-1}, \|w^i\|_1 \le B_i \right\},$$

where σ is a Lipschitz transfer function, such as the sigmoid function.

Proposition 15 *Suppose $\sigma : \mathbb{R} \to [-1,1]$ is L-Lipschitz with $\sigma(0) = 0$. Then it holds that*

$$\mathfrak{R}_T(\mathcal{F}_k) \le \left(\prod_{i=2}^{k} 16 B_i \right) L^{k-1} \left(1 + 4\sqrt{2}\log^{3/2}(eT^2)\right)^k \mathfrak{R}_T(\mathcal{F}_1).$$

In particular, for the case of

$$\mathcal{F}_1 = \left\{ x \mapsto \textstyle\sum_j w_j^1 x_j \;\middle|\; \|w\|_1 \le B_1 \right\}$$

and $\mathcal{X} \subset \mathbb{R}^d$ we have the bound

$$\mathfrak{R}_T(\mathcal{F}_k) \le \left(\prod_{i=1}^{k} 16 B_i \right) L^{k-1} \left(1 + 4\sqrt{2}\log^{3/2}(eT^2)\right)^k X_\infty \sqrt{\frac{2\log d}{T}}$$

where X_∞ is such that $\forall x \in \mathcal{X}$, $\|x\|_\infty \le X_\infty$.

Our result is a non-constructive guarantee, and, to the best of our knowledge, no algorithms for learning neural networks within the online learning model exist. It is not clear if the above bounds could be obtained via computationally efficient methods.

6.2 Margin Based Regret

In the classical statistical setting, margin bounds provide guarantees on the expected zero-one loss of a classifier based on the empirical margin zero-one error. These results form the basis of the theory of large margin classifiers (see Schapire et al., 1997; Koltchinskii and Panchenko, 2002). Recently, in the online setting, bounds of a similar flavor have been shown through the concept of margin via the Littlestone dimension (Ben-David et al., 2009). We show that our machinery can easily lead to margin bounds for binary classification problems for general function classes $\mathcal{F}$ based on their sequential Rademacher complexity. We use ideas from (Koltchinskii and Panchenko, 2002) to do this.

Proposition 16 *For any function class $\mathcal{F} \subset [-1,1]^{\mathcal{X}}$, there exists a randomized prediction strategy given by τ such that for any sequence $z_1, \ldots, z_T$ where each $z_t = (x_t, y_t) \in \mathcal{X} \times \{\pm 1\}$,*

$$\sum_{t=1}^{T} \mathbb{E}_{\hat{y}_t \sim \tau_t(z_{1:t-1})} \left[\mathbf{1}\left\{ \hat{y}_t y_t < 0 \right\} \right]$$
$$\le \inf_{\gamma > 0} \left\{ \inf_{f \in \mathcal{F}} \sum_{t=1}^{T} \mathbf{1}\left\{ f(x_t) y_t < 2\gamma \right\} + \frac{16}{\gamma} \left(1 + 4\sqrt{2}\log^{3/2}(eT^2)\right) T \mathfrak{R}_T(\mathcal{F}) + 2\sqrt{T}\left(1 + \log\log\left(\frac{1}{\gamma}\right)\right) \right\}.$$

To interpret the above bound, suppose that the sequence of y_t's is predicted with a margin 2γ by some function $f \in \mathcal{F}$. The upper bound guarantees that there exists a strategy (that does not need to know the value of γ) with cumulative loss given by the sequential Rademacher complexity of $\mathcal{F}$ divided by the margin, up to poly-logarithmic factors. Crucially, the bound does not directly depend on the dimensionality of the input space $\mathcal{X}$.

6.3 Decision Trees

We consider here the binary classification problem where the learner competes with a set of decision trees of depth no more than d. The function class $\mathcal{F}$ for this problem is defined as follows. Each $f \in \mathcal{F}$ is defined by choosing a rooted binary tree of depth no more than d and associating to each node a binary valued decision function from a set $\mathcal{H} \subseteq \{\pm 1\}^{\mathcal{X}}$. A binary value for a given x can be obtained by traversing the tree from the root according to the value of the decision function at each node and then reading off the label of the leaf. Importantly, x "reaches" only one leaf of the tree. Alternatively, for any leaf l, the membership of x is given by the conjunction

$$\prod_i \mathbf{1}\left\{h_{l,i}(x) = 1\right\}$$

where $h_{l,i}$ is either the decision function at node i along the path to the leaf l, or its negation. To complete the definition of f, we choose weights $w_l > 0$, $\sum_l w_l = 1$, along with the value $\sigma_l \in \{\pm 1\}$ of the function on each leaf l. The resulting function f can be written as

$$f(x) = \sum_l w_l \sigma_l \prod_i \mathbf{1}\left\{h_{l,i}(x) = 1\right\}$$

where the sum runs over all the leaves of the tree.

The following proposition is the online analogue of a result about decision tree learning that Bartlett and Mendelson (2003) proved in the i.i.d. setting.

Proposition 17 *Denote by $\mathcal{F}$ the class of decision trees of depth at most d with decision functions in $\mathcal{H}$. There exists a randomized strategy τ for the learner such that for any sequence of instances $z_1, \ldots, z_T$, with $z_t = (x_t, y_t) \in \mathcal{X} \times \{\pm 1\}$,*

$$\sum_{t=1}^{T} \mathbb{E}_{\hat{y}_t \sim \tau_t(z_{1:t-1})}\left[\mathbf{1}\left\{\hat{y}_t \neq y_t\right\}\right] \leq \inf_{f \in \mathcal{F}} \sum_{t=1}^{T} \mathbf{1}\left\{f(x_t) \neq y_t\right\} + \mathcal{O}\left(\sum_l \min\left(C(l), d\log^3(T)\ T\ \mathfrak{R}(\mathcal{H})\right) + \sqrt{T}\log(N)\right),$$

where $C(l)$ denotes the number of instances that reach the leaf l and are correctly classified in the decision tree f that minimizes $\sum_{t=1}^{T} \mathbf{1}\{y_t f(x_t) \leq 0\}$, with $N > 2$ being the number of leaves in this tree.

It is not clear whether computationally feasible online methods exist for learning decision trees, and this represents an interesting avenue of further research.

6.4 Transductive Learning

Let $\mathcal{F}$ be a class of functions from $\mathcal{X}$ to $\mathbb{R}$. Let

$$\widehat{\mathcal{N}}_\infty(\alpha, \mathcal{F}) = \min\left\{|G| : G \subseteq \mathbb{R}^{\mathcal{X}} \text{ s.t. } \forall f \in \mathcal{F} \;\; \exists g \in G \;\; \text{satisfying} \;\; \|f - g\|_\infty \le \alpha\right\} \tag{13}$$

be the ℓ_∞ covering number at scale α, where the cover is pointwise on all of $\mathcal{X}$. It is easy to verify that

$$\forall T, \quad \mathcal{N}_\infty(\alpha, \mathcal{F}, T) \le \widehat{\mathcal{N}}_\infty(\alpha, \mathcal{F}) \; . \tag{14}$$

Indeed, let G be a minimal cover of $\mathcal{F}$ at scale α. We claim that for any $\mathcal{X}$-valued tree of depth T, the set $V = \{\mathbf{v}^g = g \circ \mathbf{x} : g \in G\}$ of $\mathbb{R}$-valued trees is an ℓ_∞ cover of $\mathcal{F}$ on $\mathbf{x}$. Fix any $\epsilon \in \{\pm 1\}^T$ and $f \in \mathcal{F}$, and let $g \in G$ be such that $\|f - g\|_\infty \le \alpha$. Clearly $|\mathbf{v}^g_t(\epsilon) - f(\mathbf{x}_t(\epsilon))| \le \alpha$ for any $1 \le t \le T$, concluding the proof.

This simple observation can be applied in several situations. First, consider the problem of *transductive learning*, where the set $\mathcal{X} = \{x_1, \ldots, x_n\}$ is a finite set. To ensure online learnability, it is sufficient to consider an assumption on the dependence of $\widehat{\mathcal{N}}_\infty(\alpha, \mathcal{F})$ on α. An obvious example of such a class is a VC-type class with $\widehat{\mathcal{N}}_\infty(\alpha, \mathcal{F}) \le (c/\alpha)^d$ for some c which can depend on n. Assume that $\mathcal{F} \subset [-1, 1]^{\mathcal{X}}$. Substituting this bound on the covering number into (6) and choosing $\alpha = 0$, we observe that the value of the supervised game is upper bounded by $2\mathfrak{D}_T(\mathcal{F}) \le 48\sqrt{dT \log c}$ by Proposition 9. It is easy to see that if n is fixed and the problem is learnable in the batch (i.e., i.i.d.) setting, then the problem is learnable in the online transductive model.

In the transductive setting considered by Kakade and Kalai (2006), it is assumed that $n \le T$ and $\mathcal{F}$ consists of binary-valued functions. If $\mathcal{F}$ is a class with VC dimension d, the Sauer-Shelah lemma ensures that the ℓ_∞ cover is smaller than $(en/d)^d \le (eT/d)^d$. Using the previous argument with $c = eT$, we obtain a bound of $4\sqrt{dT \log(eT)}$ for the value of the game, matching the bound of Kakade and Kalai (2006) up to a constant factor.

6.5 Isotron

Kalai and Sastry (2009) introduced a method called *Isotron* for learning Single Index Models (SIM). These models generalize linear and logistic regression, generalized linear models, and classification by linear threshold functions. For brevity, we only describe the Idealized SIM problem considered by the authors. In its "batch" version, we assume that the data are revealed at once as a set $\{(x_t, y_t)\}_{t=1}^T \in \mathbb{R}^d \times \mathbb{R}$ where $y_t = u(\langle w, x_t \rangle)$ for some unknown $w \in \mathbb{R}^d$ of bounded norm and an unknown non-decreasing $u : \mathbb{R} \to \mathbb{R}$ with a bounded Lipschitz constant. Given this data, the goal is to iteratively find the function u and the direction w, making as few mistakes as possible. The error is measured as $\frac{1}{T}\sum_{t=1}^T (f_i(x_t) - y_t)^2$, where $f_i(x) = u_i(\langle w_i, x \rangle)$ is the iterative approximation found by the algorithm on the ith round. The elegant computationally efficient method presented by Kalai and Sastry (2009) is motivated by Perceptron, and a natural open question posed by the authors is whether there is an online variant of Isotron. Before even attempting a quest for such an algorithm, we can ask a more basic question: is the (Idealized) SIM problem even learnable in the online framework? After all, most online methods deal with convex functions, but u is only assumed to be Lipschitz and non-decreasing. We answer the question easily with the tools we have developed.

We are interested in online learnability of

$$\mathcal{H} = \left\{ f(x,y) = (y - u(\langle w, x\rangle))^2 \mid u : [-1,1] \to [-1,1] \text{ 1-Lipschitz }, \ \|w\|_2 \le 1 \right\} \tag{15}$$

in the supervised setting, over $\mathcal{X} = B_2$ (the unit Euclidean ball in $\mathbb{R}^d$) and $\mathcal{Y} = [-1,1]$. In particular, we prove the result for Lipschitz, but not necessarily non-decreasing functions. It is evident that $\mathcal{H}$ is a composition with three levels: the squared loss, the Lipschitz non-decreasing function, and the linear function. The proof of the following proposition shows that the covering number of the class does not increase much under these compositions.

Proposition 18 *The class $\mathcal{H}$ defined in* (15) *is online learnable in the (improper) supervised learning setting. Moreover, the minimax regret is*

$$\mathcal{O}(\sqrt{T}\log^{3/2}(T)).$$

Once again, it is not clear whether a computationally efficient method attaining the above guarantee exists.

6.6 Prediction of Individual Sequences with Static Experts

We also consider the problem of prediction of individual sequences, which has been studied both in information theory and in learning theory. In particular, in the case of binary prediction, Cesa-Bianchi and Lugosi (1999) proved upper bounds on the minimax value in terms of the (classical) Rademacher complexity and the (classical) Dudley integral. One of the assumptions made by Cesa-Bianchi and Lugosi (1999) is that experts are *static*. That is, their prediction only depends on the current round, not on the past information. Formally, we define static experts as vectors $\bar{f} = (f_1, \dots, f_T) \in [0,1]^T$, and let $\mathcal{F}$ denote a class of such experts. Let $\mathcal{Y} = \{0,1\}$, putting us in the scenario of binary classification with no side information. Then regret on a particular sequence $y_1, \dots, y_T$ can be written as

$$\sum_{t=1}^{T} \ell_t(\bar{f}_t, y_t) - \inf_{\bar{f}\in\mathcal{F}} \sum_{t=1} \ell_t(\bar{f}, y_t),$$

where $\bar{f}_t$ is the expert chosen by the learning algorithm at time t. Observe that the proof of Theorem 7 does not require the loss to be time independent. In the case of absolute loss, the Rademacher complexity appearing on the right hand side in Theorem 7 becomes

$$\sup_{\mathbf{y}} \mathbb{E}_\epsilon \left[\sup_{\bar{f}\in\mathcal{F}} \sum_{t=1}^{T} \epsilon_t \ell_t(\bar{f}, \mathbf{y}_t(\epsilon)) \right] = \sup_{\mathbf{y}} \mathbb{E}_\epsilon \left[\sup_{\bar{f}\in\mathcal{F}} \sum_{t=1}^{T} \epsilon_t |f_t - \mathbf{y}_t(\epsilon)| \right] .$$

where the supremum is over all $\mathcal{Y}$-valued trees of depth T. Noting that for $f \in [0,1], y \in \{0,1\}$, $|f-y|$ can be written as $(1-2y)f + y$, the above equals

$$\sup_{\mathbf{y}} \mathbb{E}_\epsilon \left[\left(\sup_{\bar{f}\in\mathcal{F}} \sum_{t=1}^{T} \epsilon_t (1 - 2\mathbf{y}_t(\epsilon)) f_t \right) + \sum_{t=1}^{T} \epsilon_t \mathbf{y}_t(\epsilon) \right] = \sup_{\mathbf{y}} \mathbb{E}_\epsilon \left[\sup_{\bar{f}\in\mathcal{F}} \sum_{t=1}^{T} \epsilon_t (1 - 2\mathbf{y}_t(\epsilon)) f_t \right].$$

It can be easily verified that the joint distribution of $\{\epsilon_t(1-2\mathbf{y}_t(\epsilon))\}_{t=1}^T$ is still i.i.d. Rademacher and hence the value of the game is upper bounded by

$$2\mathbb{E}_\epsilon\left[\sup_{\bar{f}\in\mathcal{F}}\sum_{t=1}^T \epsilon_t f_t\right],$$

recovering the upper bound of Theorem 3 in (Cesa-Bianchi and Lugosi, 1999). We note that for this particular scenario, the factor of 2 (that appears because of symmetrization) is not needed. This factor is the price we pay for deducing the result from the general statement of Theorem 7.

7. Discussion

The tools provided in this paper allow us to establish existence of regret minimization algorithms by working directly with the minimax value. The non-constructive nature of our results is due to the application of the minimax theorem: the dual strategy does not give a handle on the primal strategy. Furthermore, by passing to upper bounds on the dual formulation (2) of the value of the game, we remove the dependence on the dual strategy altogether. After the original paper (Rakhlin et al., 2010) appeared, the algorithmic approach has been developed by Rakhlin et al. (2012) who showed that the prediction for round t can be obtained by appealing to the minimax theorem for rounds $t+1$ to T, yet keeping the minimax expression for round t as is. The notion of a relaxation (in the spirit of approximate dynamic programming) then allowed the authors to develop a general recipe for deriving computationally feasible prediction methods. The techniques of the present paper form the basis for the algorithmic developments of Rakhlin et al. (2012). We refer the reader to (Rakhlin and Sridharan, 2014; Rakhlin et al., 2012) for details.

Acknowledgments

We would like to thank J. Michael Steele and Dean Foster for helpful discussions. We gratefully acknowledge the support of NSF under grants CAREER DMS-0954737 and CCF-1116928.

Appendix A. A Minimax Theorem

The minimax theorem is one of this paper's main workhorses. For completeness, we state a general version of this theorem — the von Neumann-Fan minimax theorem — due to Borwein (2014) (see also Borwein and Zhuang, 1986).

Theorem 19 (Borwein, 2014) *Let $\mathcal{A}$ and $\mathcal{B}$ be Banach spaces. Let $A \subset \mathcal{A}$ be nonempty, weakly compact, and convex, and let $B \subset \mathcal{B}$ be nonempty and convex. Let $g : \mathcal{A}\times\mathcal{B} \to \mathbb{R}$ be concave with respect to $b \in B$ and convex and lower-semicontinuous with respect to $a \in A$, and weakly continuous in a when restricted to A. Then*

$$\sup_{b\in B}\inf_{a\in A} g(a,b) = \inf_{a\in A}\sup_{b\in B} g(a,b). \tag{16}$$

In the proof of Theorem 1, the minimax theorem is invoked to assure that

$$\inf_{q_t \in \mathcal{Q}} \sup_{p_t \in \mathcal{P}} \mathbb{E}\left[\ell(f_t, z_t) + \xi(z_t)\right] = \sup_{p_t \in \mathcal{P}} \inf_{q_t \in \mathcal{Q}} \mathbb{E}\left[\ell(f_t, z_t) + \xi(z_t)\right], \tag{17}$$

where $\xi(z_t)$ is a rather complicated function that includes the repeated infima and suprema from steps $t+1$ to T of regret expression that includes the variable z_t (but not f_t). The expectation in (17) is with respect to $f_t \sim q_t$ and $z_t \sim p_t$. To apply (16), we take g to be the bilinear form in q_t and p_t, with $A = \mathcal{Q}$ and $B = \mathcal{P}$. Equipped with the total variation distance, $\mathcal{Q}$ and $\mathcal{P}$ can be seen as subsets of a Banach space of measures on $\mathcal{F}$ and $\mathcal{Z}$, respectively. In terms of conditions, it is enough to check weak compactness of $\mathcal{Q}$ and assume continuity of the loss function (lower semi-continuity can be used as well).

Weak compactness of the set of probability measures on a complete separable metric space is equivalent to uniform tightness by the fundamental result of Prohorov (see, e.g., Bogachev 2007, Theorem 8.6.2., and van der Vaart and Wellner 1996). If $\mathcal{F}$ itself is compact, then the set $\Delta(\mathcal{F})$ of probability measures on $\mathcal{F}$ is tight, and hence (under the continuity of the loss) the minimax theorem holds. If $\mathcal{F}$ is not compact, tightness can be established under the following general condition. According to Example 8.6.5 (ii) in Bogachev (2007), a family $\Delta(\mathcal{F})$ of Borel probability measures on a separable *reflexive* Banach space E is uniformly tight (under the weak topology) precisely when there exists a function $V : E \to [0, \infty)$ continuous in the norm topology such that

$$\lim_{\|f\| \to \infty} V(f) = \infty \quad \text{and} \quad \sup_{q \in \Delta(\mathcal{F})} \mathbb{E}_{f \sim q} V(f) < \infty.$$

As an example, if $\mathcal{F}$ is a subset of a ball in E, it is enough to take $V(f) = \|f\|$.

Finally, we remark that in the supervised learning case by considering the improper learning scenario we allow x_t to be observed before the choice $\hat{y}_t$ is made. Therefore, we do not need to invoke the minimax theorem on the space of functions $\mathcal{F}$, but rather (see the proof of Theorem 8) for two real-valued decisions in a bounded interval. This makes the application of the minimax theorem straightforward.

Appendix B. Proofs

Proof [**of Theorem 1**] For brevity, denote $\psi(z_{1:T}) = \inf_{f \in \mathcal{F}} \sum_{t=1}^{T} \ell(f, z_t)$. The first step in the proof is to appeal to the minimax theorem for every couple of inf and sup:

$$\begin{aligned}
\mathcal{V}_T(\mathcal{F}) &= \inf_{q_1} \sup_{p_1} \mathbb{E}_{\substack{f_1 \sim q_1 \\ z_1 \sim p_1}} \dots \inf_{q_T} \sup_{p_T} \mathbb{E}_{\substack{f_T \sim q_T \\ z_T \sim p_T}} \left\{ \sum_{t=1}^{T} \ell(f_t, z_t) - \psi(z_{1:T}) \right\} \\
&= \sup_{p_1} \inf_{q_1} \mathbb{E}_{\substack{f_1 \sim q_1 \\ z_1 \sim p_1}} \dots \sup_{p_T} \inf_{q_T} \mathbb{E}_{\substack{f_T \sim q_T \\ z_T \sim p_T}} \left\{ \sum_{t=1}^{T} \ell(f_t, z_t) - \psi(z_{1:T}) \right\} \\
&= \sup_{p_1} \inf_{f_1} \mathbb{E}_{z_1 \sim p_1} \dots \sup_{p_T} \inf_{f_T} \mathbb{E}_{z_T \sim p_T} \left\{ \sum_{t=1}^{T} \ell(f_t, z_t) - \psi(z_{1:T}) \right\},
\end{aligned}$$

where q_t and p_t range over $\mathcal{Q}$ and $\mathcal{P}$, the sets of distributions on $\mathcal{F}$ and $\mathcal{Z}$, respectively. From now on, it will be understood that z_t has distribution p_t. By moving the expectation

with respect to z_T and then the infimum with respect to f_T inside the expression, we arrive at

$$\sup_{p_1}\inf_{f_1}\mathbb{E}_{z_1}\ldots\sup_{p_{T-1}}\inf_{f_{T-1}}\mathbb{E}_{z_{T-1}}\sup_{p_T}\left\{\sum_{t=1}^{T-1}\ell(f_t,z_t)+\left[\inf_{f_T}\mathbb{E}_{z_T}\ell(f_T,z_T)\right]-\mathbb{E}_{z_T}\psi(z_{1:T})\right\}$$
$$=\sup_{p_1}\inf_{f_1}\mathbb{E}_{z_1}\ldots\sup_{p_{T-1}}\inf_{f_{T-1}}\mathbb{E}_{z_{T-1}}\sup_{p_T}\mathbb{E}_{z_T}\left\{\sum_{t=1}^{T-1}\ell(f_t,z_t)+\left[\inf_{f_T}\mathbb{E}_{z_T}\ell(f_T,z_T)\right]-\psi(z_{1:T})\right\}. \tag{18}$$

Let us now repeat the procedure for step $T-1$. The above expression is equal to

$$\sup_{p_1}\inf_{f_1}\mathbb{E}_{z_1}\ldots\sup_{p_{T-1}}\inf_{f_{T-1}}\mathbb{E}_{z_{T-1}}\left\{\sum_{t=1}^{T-1}\ell(f_t,z_t)+\sup_{p_T}\mathbb{E}_{z_T}\left[\inf_{f_T}\mathbb{E}_{z_T}\ell(f_T,z_T)-\psi(z_{1:T})\right]\right\}$$

which, in turn, is equal to

$$\sup_{p_1}\inf_{f_1}\mathbb{E}_{z_1}\ldots\sup_{p_{T-1}}\left\{\sum_{t=1}^{T-2}\ell(f_t,z_t)+\left[\inf_{f_{T-1}}\mathbb{E}_{z_{T-1}}\ell(f_{T-1},z_{T-1})\right]\right.$$
$$\left.+\mathbb{E}_{z_{T-1}}\sup_{p_T}\mathbb{E}_{z_T}\left[\inf_{f_T}\mathbb{E}_{z_T}\ell(f_T,z_T)-\psi(z_{1:T})\right]\right\}$$
$$=\sup_{p_1}\inf_{f_1}\mathbb{E}_{z_1}\ldots\sup_{p_{T-1}}\mathbb{E}_{z_{T-1}}\sup_{p_T}\mathbb{E}_{z_T}\left\{\sum_{t=1}^{T-2}\ell(f_t,z_t)+\left[\inf_{f_{T-1}}\mathbb{E}_{z_{T-1}}\ell(f_{T-1},z_{T-1})\right]\right.$$
$$\left.+\left[\inf_{f_T}\mathbb{E}_{z_T}\ell(f_T,z_T)\right]-\psi(z_{1:T})\right\}.$$

Continuing in this fashion for $T-2$ and all the way down to $t=1$ proves the theorem. ■

Proof [**of Lemma 4**] Without loss of generality assume that the Lipschitz constant $L=1$, as the general case follows by scaling ϕ. Fix a $\mathcal{Z}$-valued tree $\mathbf{z}$ of depth T. We first claim that

$$\log\,\mathcal{N}_2(\beta,\phi\circ\mathcal{G},\mathbf{z})\le\sum_{j=1}^{k}\log\,\mathcal{N}_\infty(\beta,\mathcal{G}_j,\mathbf{z})\ .$$

Suppose $V_1,\ldots,V_k$ are minimal β-covers with respect to ℓ_∞ for $\mathcal{G}_1,\ldots,\mathcal{G}_k$ on the tree $\mathbf{z}$. Consider the set

$$V^\phi=\{\mathbf{v}^\phi\,:\,\mathbf{v}\in V_1\times\ldots\times V_k\},$$

where $\mathbf{v}^\phi$ is the tree such that $\mathbf{v}_t^\phi(\epsilon)=\phi(\mathbf{v}_t(\epsilon),\mathbf{z}_t(\epsilon))$. Then, for any $g=(g_1,\ldots,g_k)\in\mathcal{G}$ and any $\epsilon\in\{\pm1\}^T$, with representatives $(\mathbf{v}^1,\ldots,\mathbf{v}^k)\in V_1\times\ldots\times V_k$, we have,

$$\sqrt{\frac{1}{T}\sum_{t=1}^{T}\left(\phi(g(\mathbf{z}_t(\epsilon)),\mathbf{z}_t(\epsilon))-\mathbf{v}_t^\phi(\epsilon)\right)^2}\le\max_{t\in[T]}\left|\phi(g(\mathbf{z}_t(\epsilon)),\mathbf{z}_t(\epsilon))-\mathbf{v}_t^\phi(\epsilon)\right|$$
$$=\max_{t\in[T]}|\phi(g(\mathbf{z}_t(\epsilon)),\mathbf{z}_t(\epsilon))-\phi(\mathbf{v}_t(\epsilon),\mathbf{z}_t(\epsilon))|\le\max_{j\in[k]}\max_{t\in[T]}|g_j(\mathbf{z}_t(\epsilon)))-\mathbf{v}_t^j(\epsilon)|\le\beta.$$

Thus we see that V^ϕ is an β-cover with respect to ℓ_∞ for $\phi \circ \mathcal{G}$ on $\mathbf{z}$. Hence

$$\log\ \mathcal{N}_2(\beta, \phi \circ \mathcal{G}, \mathbf{z}) \leq \log(|V^\phi|) = \sum_{j=1}^{k} \log(|V_j|) = \sum_{j=1}^{k} \log\ \mathcal{N}_\infty(\beta, \mathcal{G}_j, \mathbf{z}). \tag{19}$$

For any $g \in \mathcal{G}$ and $z \in \mathcal{Z}$, the value $\phi(g(z), z)$ is contained in the interval $[-1 + \phi(\mathbf{0}, z), +1 + \phi(\mathbf{0}, z)]$ by the Lipschitz property. Consider the $\mathbb{R}$-valued tree $\phi(\mathbf{0}, \cdot) \circ \mathbf{z}$. We now center by this tree and consider the set of trees

$$\{\phi(g(\cdot), \cdot) \circ \mathbf{z} - \phi(\mathbf{0}, \cdot) \circ \mathbf{z} : g \in \mathcal{G}\}.$$

The centering does not change the size of the cover calculated in (19), but allows us to invoke (7) since the function values are now in $[-1, 1]$:

$$\begin{aligned}\mathfrak{R}_T(\phi \circ \mathcal{G}, \mathbf{z}) &\leq \inf_\alpha \left\{ 4\alpha + \frac{12}{\sqrt{T}} \int_\alpha^1 \sqrt{\sum_{j=1}^{k} \log\ \mathcal{N}_\infty(\beta, \mathcal{G}_j, \mathbf{z})}\ d\beta \right\} \\ &\leq \inf_\alpha \left\{ 4\alpha + \frac{12}{\sqrt{T}} \sum_{j=1}^{k} \int_\alpha^1 \sqrt{\log\ \mathcal{N}_\infty(\beta, \mathcal{G}_j, \mathbf{z})}\, d\beta \right\}. \end{aligned} \tag{20}$$

We substitute the upper bound on covering numbers in (8) for each $\mathcal{G}_j$ and arrive at an upper bound of

$$\inf_\alpha \left\{ 4\alpha + \frac{12}{\sqrt{T}} \sum_{j=1}^{k} \int_\alpha^1 \sqrt{\mathrm{fat}_\beta(\mathcal{G}_j) \log(2eT/\beta)} d\beta \right\}. \tag{21}$$

Lemma 2 of Rakhlin et al. (2014) implies that for any $\beta > 2\mathfrak{R}_T(\mathcal{G}_j)$,

$$\mathrm{fat}_\beta(\mathcal{G}_j) \leq \frac{32T\ \mathfrak{R}_T(\mathcal{G}_j)^2}{\beta^2}\ .$$

Let $j^* = \operatorname{argmax}_j \mathfrak{R}_T(\mathcal{G}_j)$. Substituting this together with the value of $\alpha = 2\mathfrak{R}_T(\mathcal{G}_{j^*})$ into (21) yields an upper bound

$$8\ \mathfrak{R}_T(\mathcal{G}_{j^*}) + 48\sqrt{2}\ \sum_{j=1}^{k} \mathfrak{R}_T(\mathcal{G}_j) \int_{2\mathfrak{R}_T(\mathcal{G}_{j^*})}^1 \frac{1}{\beta} \sqrt{\log(2eT/\beta)} d\beta.$$

Using the fact that for any $b > 1$ and $\alpha \in (0, 1)$

$$\int_\alpha^1 \frac{1}{\beta} \sqrt{\log(b/\beta)} d\beta = \int_b^{b/\alpha} \frac{1}{x} \sqrt{\log x} dx = \frac{2}{3} \log^{3/2}(x) \Big|_b^{b/\alpha} \leq \frac{2}{3} \log^{3/2}(b/\alpha) \tag{22}$$

we obtain a further upper bound of

$$8\ \mathfrak{R}_T(\mathcal{G}_{j^*}) + 32\sqrt{2}\ \sum_{j=1}^{k} \mathfrak{R}_T(\mathcal{G}_j)\ \log^{3/2}\left(\frac{eT}{\mathfrak{R}_T(\mathcal{G}_{j^*})}\right)\ .$$

Replacing the first term by $8\sum_j \mathfrak{R}_T(\mathcal{G}_j)$, we conclude that

$$\mathfrak{R}_T(\phi\circ\mathcal{G},\mathbf{z}) \le 8\left(1+4\sqrt{2}\log^{3/2}(eT^2)\right)\sum_{j=1}^{k}\mathfrak{R}_T(\mathcal{G}_j)$$

as long as $\mathfrak{R}_T(\mathcal{G}_j)\ge 1/T$ for each j. The statement is concluded by observing that $\mathbf{z}$ was chosen arbitrarily. ■

Proof [**of Corollary 6**] We first extend the binary function b to a function $\bar{b}$ to any $x\in\mathbb{R}^k$ as follows :

$$\bar{b}(x) = \begin{cases} (1-\|x-a\|_\infty)b(a) & \text{if } \|x-a\|_\infty < 1 \text{ for some } a\in\{\pm1\}^k \\ 0 & \text{otherwise} \end{cases}$$

First note that $\bar{b}$ is well-defined since all points in the k-cube are separated by L_∞ distance 2. Further note that $\bar{b}$ is 1-Lipschitz w.r.t. the L_∞ norm and so applying Lemma 4 we conclude the statement of the corollary. ■

Proof [**of Theorem 7**] Let $\mathbb{E}_{t-1}[\cdot]=\mathbb{E}[\cdot|Z_1,\dots,Z_{t-1}]$ denote the conditional expectation. Using Theorem 1 we have,

$$\begin{aligned}
\mathcal{V}_T(\mathcal{F}) &= \sup_{p_1}\mathop{\mathbb{E}}_{Z_1\sim p_1}\dots\sup_{p_T}\mathop{\mathbb{E}}_{Z_T\sim p_T}\left[\sum_{t=1}^{T}\inf_{f_t\in\mathcal{F}}\mathbb{E}_{t-1}\ell(f_t,\cdot)-\inf_{f\in\mathcal{F}}\sum_{t=1}^{T}\ell(f,Z_t)\right] \\
&= \sup_{p_1}\mathop{\mathbb{E}}_{Z_1\sim p_1}\dots\sup_{p_T}\mathop{\mathbb{E}}_{Z_T\sim p_T}\left[\sup_{f\in\mathcal{F}}\left\{\sum_{t=1}^{T}\inf_{f_t\in\mathcal{F}}\mathbb{E}_{t-1}\ell(f_t,\cdot)-\sum_{t=1}^{T}\ell(f,Z_t)\right\}\right] \\
&\le \sup_{p_1}\mathop{\mathbb{E}}_{Z_1\sim p_1}\dots\sup_{p_T}\mathop{\mathbb{E}}_{Z_T\sim p_T}\left[\sup_{f\in\mathcal{F}}\left\{\sum_{t=1}^{T}\mathbb{E}_{t-1}\ell(f,\cdot)-\sum_{t=1}^{T}\ell(f,Z_t)\right\}\right]. \qquad (23)
\end{aligned}$$

The upper bound is obtained by replacing each infimum by a particular choice f. This step also holds if the choice f_t of the learner comes from a larger set $\mathcal{G}$, as long as $\mathcal{F}\subseteq\mathcal{G}$. The proof is concluded by appealing to (3).

■

Proof [**of Theorem 8**]

Let $\tilde{Q}$ denote the set of distributions on $\mathcal{Y}=[-1,1]$. By convexity,

$$\sum_{t=1}^{T}\ell(\hat{y}_t,y_t)-\inf_{f\in\mathcal{F}}\sum_{t=1}^{T}\ell(f(x_t),y_t)\le\sup_{f\in\mathcal{F}}\sum_{t=1}^{T}\ell'(\hat{y}_t,y_t)\left(\hat{y}_t-f(x_t)\right),$$

where $\ell'(\hat{y}_t,y_t)$ is a subgradient of the function $y\mapsto\ell(\cdot,y_t)$ at $\hat{y}_t$. Then the minimax value (10) can be upper bounded as

$$\mathcal{V}_T^S(\mathcal{F})\le\sup_{x_1}\inf_{q_1\in\tilde{Q}}\sup_{y_1}\mathop{\mathbb{E}}_{\hat{y}_1\sim q_1}\dots\sup_{x_T}\inf_{q_T\in\tilde{Q}}\sup_{y_T}\mathbb{E}_{\hat{y}_T\sim q_T}\left[\sup_{f\in\mathcal{F}}\sum_{t=1}^{T}\ell'(\hat{y}_t,y_t)\left(\hat{y}_t-f(x_t)\right)\right].$$

By the Lipschitz property of ℓ, we can replace each subgradient $\ell'(\hat{y}_t, y_t)$ with a number $s_t \in [-L, L]$ to obtain the upper bound

$$\sup_{x_1} \inf_{q_1 \in \tilde{Q}} \sup_{y_1} \mathop{\mathbb{E}}_{\hat{y}_1 \sim q_1} \sup_{s_1 \in [-L,L]} \dots \sup_{x_T} \inf_{q_T \in \tilde{Q}} \sup_{y_T} \mathop{\mathbb{E}}_{\hat{y}_T \sim q_T} \sup_{s_T \in [-L,L]} \left\{ \sup_{f \in \mathcal{F}} \sum_{t=1}^{T} s_t \left(\hat{y}_t - f(x_t) \right) \right\}.$$

Since y_t's no longer appear in the optimization objective, we can simply write the above as

$$\begin{aligned} &\sup_{x_1} \inf_{q_1 \in \tilde{Q}} \mathop{\mathbb{E}}_{\hat{y}_1 \sim q_1} \sup_{s_1 \in [-L,L]} \dots \sup_{x_T} \inf_{q_T \in \tilde{Q}} \mathop{\mathbb{E}}_{\hat{y}_T \sim q_T} \sup_{s_T \in [-L,L]} \left\{ \sup_{f \in \mathcal{F}} \sum_{t=1}^{T} s_t \left(\hat{y}_t - f(x_t) \right) \right\} \\ &= \sup_{x_1} \inf_{\hat{y}_1 \in [-1,1]} \sup_{s_1 \in [-L,L]} \dots \sup_{x_T} \inf_{\hat{y}_T \in [-1,1]} \sup_{s_T \in [-L,L]} \left\{ \sup_{f \in \mathcal{F}} \sum_{t=1}^{T} s_t \left(\hat{y}_t - f(x_t) \right) \right\}, \end{aligned}$$

where the equality follows because infima are obtained at point distributions. By the same reasoning, we now pass to distributions over s_t's:

$$\sup_{x_1} \inf_{\hat{y}_1 \in [-1,1]} \sup_{p_1} \mathop{\mathbb{E}}_{s_1 \sim p_1} \dots \sup_{x_T} \inf_{\hat{y}_T \in [-1,1]} \sup_{p_T} \mathbb{E}_{s_T \sim p_T} \left[\sum_{t=1}^{T} s_t \cdot \hat{y}_t - \inf_{f \in \mathcal{F}} \sum_{t=1}^{T} s_t f(x_t) \right]. \tag{24}$$

From now on, it will be understood that the supremum over p_t ranges over all distributions supported on $[-L, L]$, for any t, and s_t has distribution p_t. Now note that

$$\mathbb{E}_{s_T} \left[\sum_{t=1}^{T} s_t \cdot \hat{y}_t - \inf_{f \in \mathcal{F}} \sum_{t=1}^{T} s_t \cdot f(x_t) \right]$$

is concave (linear) in p_T and is convex in $\hat{y}_T$ and hence by the minimax theorem,

$$\begin{aligned} &\inf_{\hat{y}_T \in [-1,1]} \sup_{p_T} \mathbb{E}_{s_T} \left[\sum_{t=1}^{T} s_t \cdot \hat{y}_t - \inf_{f \in \mathcal{F}} \sum_{t=1}^{T} s_t f(x_t) \right] = \sup_{p_T} \inf_{\hat{y}_T \in [-1,1]} \mathbb{E}_{s_T} \left[\sum_{t=1}^{T} s_t \cdot \hat{y}_t - \inf_{f \in \mathcal{F}} \sum_{t=1}^{T} s_t f(x_t) \right] \\ &= \sum_{t=1}^{T-1} s_t \cdot \hat{y}_t + \sup_{p_T} \mathbb{E}_{s_T} \left[\inf_{\hat{y}_T \in [-1,1]} \mathbb{E}_{s_T} [s_T] \cdot \hat{y}_T - \inf_{f \in \mathcal{F}} \sum_{t=1}^{T} s_t f(x_t) \right], \end{aligned}$$

where the last step is similar to the one in the proof of Theorem 1, specifically (18). Similarly note that the term

$$\mathbb{E}_{s_{T-1}} \left[\sum_{t=1}^{T-1} s_t \cdot \hat{y}_t + \sup_{p_T, x_T} \mathbb{E}_{s_T} \left[\inf_{\hat{y}_T \in [-1,1]} \mathbb{E}_{s_T} [s_T] \cdot \hat{y}_T - \inf_{f \in \mathcal{F}} \sum_{t=1}^{T} s_t f(x_t) \right] \right]$$

is concave (linear) in p_{T-1} and is convex in $\hat{y}_{T-1}$ and hence again by the minimax theorem,

$$\begin{aligned} &\inf_{\hat{y}_{T-1} \in [-1,1]} \sup_{p_{T-1}} \mathbb{E}_{s_{T-1}} \left[\sum_{t=1}^{T-1} s_t \cdot \hat{y}_t + \sup_{p_T, x_T} \mathbb{E}_{s_T} \left[\inf_{\hat{y}_T \in [-1,1]} \mathbb{E}_{s_T} [s_T] \cdot \hat{y}_T - \inf_{f \in \mathcal{F}} \sum_{t=1}^{T} s_t f(x_t) \right] \right] \\ &= \sup_{p_{T-1}} \inf_{\hat{y}_{T-1} \in [-1,1]} \mathbb{E}_{s_{T-1}} \left[\sum_{t=1}^{T-1} s_t \cdot \hat{y}_t + \sup_{p_T, x_T} \mathbb{E}_{s_T} \left[\inf_{\hat{y}_T \in [-1,1]} \mathbb{E}_{s_T} [s_T] \cdot \hat{y}_T - \inf_{f \in \mathcal{F}} \sum_{t=1}^{T} s_t f(x_t) \right] \right] \\ &= \sum_{t=1}^{T-2} s_t \cdot \hat{y}_t + \sup_{p_{T-1}} \mathop{\mathbb{E}}_{s_{T-1}} \sup_{p_T, x_T} \mathbb{E}_{s_T} \left[\sum_{t=T-1}^{T} \inf_{\hat{y}_t \in [-1,1]} \mathbb{E}_{s_t} [s_t] \cdot \hat{y}_t - \inf_{f \in \mathcal{F}} \sum_{t=1}^{T} s_t f(x_t) \right]. \end{aligned}$$

Proceeding in similar fashion and using this in (24) we conclude that,

$$\begin{aligned}
\mathcal{V}_T^S(\mathcal{F}) &\le \sup_{x_1} \inf_{\hat{y}_1 \in [-1,1]} \sup_{p_1} \mathop{\mathbb{E}}_{s_1 \sim p_1} \dots \sup_{x_T} \inf_{\hat{y}_T \in [-1,1]} \sup_{p_T} \mathbb{E}_{s_T \sim p_T} \left[\sum_{t=1}^T s_t \cdot \hat{y}_t - \inf_{f \in \mathcal{F}} \sum_{t=1}^T s_t f(x_t) \right] \\
&= \sup_{x_1} \sup_{p_1} \mathop{\mathbb{E}}_{s_1 \sim p_1} \dots \sup_{x_T} \sup_{p_T} \mathop{\mathbb{E}}_{s_T \sim p_T} \left[\sum_{t=1}^T \inf_{\hat{y}_t \in [-1,1]} \mathbb{E}_{s_t \sim p_t}[s_t] \cdot \hat{y}_t - \inf_{f \in \mathcal{F}} \sum_{t=1}^T s_t f(x_t) \right] \\
&\le \sup_{x_1} \sup_{p_1} \mathop{\mathbb{E}}_{s_1 \sim p_1} \dots \sup_{x_T} \sup_{p_T} \mathbb{E}_{s_T \sim p_T} \left[\sup_{f \in \mathcal{F}} \sum_{t=1}^T \left(\mathbb{E}_{s_t \sim p_t}[s_t] - s_t \right) f(x_t) \right],
\end{aligned}$$

where we replaced each $\hat{y}_t$ with a potentially suboptimal choice $f(x_t)$. Passing the expectation past the suprema we obtain an upper bound

$$\sup_{x_1} \sup_{p_1} \mathop{\mathbb{E}}_{s_1, s_1' \sim p_1} \dots \sup_{x_T} \sup_{p_T} \mathbb{E}_{s_T, s_T' \sim p_T} \left[\sup_{f \in \mathcal{F}} \sum_{t=1}^T \left(s_t' - s_t \right) f(x_t) \right] \tag{25}$$

$$= \sup_{x_1} \sup_{p_1} \mathop{\mathbb{E}}_{s_1, s_1' \sim p_1} \mathop{\mathbb{E}}_{\epsilon_1} \dots \sup_{x_T} \sup_{p_T} \mathop{\mathbb{E}}_{s_T, s_T' \sim p_T} \mathbb{E}_{\epsilon_T} \left[\sup_{f \in \mathcal{F}} \sum_{t=1}^T \epsilon_t \left(s_t' - s_t \right) f(x_t) \right]$$

$$\le \sup_{x_1} \sup_{s_1 \in [-2L, 2L]} \mathop{\mathbb{E}}_{\epsilon_1} \dots \sup_{x_T} \sup_{s_T \in [-2L, 2L]} \mathbb{E}_{\epsilon_T} \left[\sup_{f \in \mathcal{F}} \sum_{t=1}^T \epsilon_t s_t f(x_t) \right]$$

$$= \sup_{x_1} \sup_{s_1 \in \{-2L, 2L\}} \mathop{\mathbb{E}}_{\epsilon_1} \dots \sup_{x_T} \sup_{s_T \in \{-2L, 2L\}} \mathbb{E}_{\epsilon_T} \left[\sup_{f \in \mathcal{F}} \sum_{t=1}^T \epsilon_t s_t f(x_t) \right] \tag{26}$$

$$= 2L \sup_{x_1} \sup_{s_1 \in \{-1,1\}} \mathop{\mathbb{E}}_{\epsilon_1} \dots \sup_{x_T} \sup_{s_T \in \{-1,1\}} \mathbb{E}_{\epsilon_T} \left[\sup_{f \in \mathcal{F}} \sum_{t=1}^T \epsilon_t s_t f(x_t) \right], \tag{27}$$

where the last inequality is because, for every $t \in [T]$, we have convexity in s_t and so supremum is achieved at either $-2L$ or $2L$. Notice that after using convexity to go to gradients, the proof technique above basically mimics the proofs of Theorems 1 and 7 to get to a symmetrized term as we did in those theorems. Now consider any arbitrary function $\psi : \{\pm 1\} \mapsto \mathbb{R}$, we have that

$$\sup_{s \in \{\pm 1\}} \mathbb{E}_\epsilon [\psi(s \cdot \epsilon)] = \sup_{s \in \{\pm 1\}} \frac{1}{2} \left(\psi(+s) + \psi(-s) \right) = \frac{1}{2} \left(\psi(+1) + \psi(-1) \right) = \mathbb{E}_\epsilon [\psi(\epsilon)] .$$

Since in (27), for each t, s_t and ϵ_t appear together as $\epsilon_t \cdot s_t$ using the above equation repeatedly, we conclude that

$$\begin{aligned}
\mathcal{V}_T^S(\mathcal{F}) &\le 2L \sup_{x_1} \sup_{s_1 \in \{-1,1\}} \mathbb{E}_{\epsilon_1} \dots \sup_{x_T} \sup_{s_T \in \{-1,1\}} \mathbb{E}_{\epsilon_T} \left[\sup_{f \in \mathcal{F}} \sum_{t=1}^T \epsilon_t s_t f(x_t) \right] \\
&= 2L \sup_{x_1} \mathbb{E}_{\epsilon_1} \dots \sup_{x_T} \mathbb{E}_{\epsilon_T} \left[\sup_{f \in \mathcal{F}} \sum_{t=1}^T \epsilon_t f(x_t) \right].
\end{aligned} \tag{28}$$

We now claim that the above supremum can be written in terms of an $\mathcal{X}$-valued tree. Briefly, the solution for x_1 in (28) is attained (for simplicity, assume the supremum is attained) at

an optimal value x_1^*. The optimal value x_2^* can be calculated for $\epsilon_1 = 1$ and $\epsilon_1 = -1$. Arguing in this manner leads to a tree $\mathbf{x}$. We conclude

$$\mathcal{V}_T^S(\mathcal{F}) \le 2L \sup_{\mathbf{x}} \mathbb{E}_{\epsilon_{1:T}}\left[\sup_{f\in\mathcal{F}} \sum_{t=1}^{T} \epsilon_t f(\mathbf{x}_t(\epsilon))\right] = 2\,L\,T\,\mathfrak{R}_T(\mathcal{F}).$$

■

Proof [**of Proposition 9**] For the upper bound, we start by using Theorem 8 for absolute loss, which has a Lipschitz constant of 1, to bound the value of the game by sequential Rademacher complexity,

$$\frac{1}{T}\mathcal{V}_T^{\mathrm{S}}(\mathcal{F}) \le 2\,\mathfrak{R}_T(\mathcal{F})\ .$$

We combine the above inequality with (7) and (8) to obtain the upper bound.

Observe that a lower bound on the value can be obtained by choosing any particular joint distribution on sequences $(x_1, y_1), \ldots, (x_t, y_t)$ in (2):

$$\mathcal{V}_T^{\mathrm{S}}(\mathcal{F}) \ge \mathbb{E}\left[\sum_{t=1}^{T} \inf_{f_t\in\mathcal{F}} \mathbb{E}_{(x_t,y_t)}\left[|y_t - f_t(x_t)| \;\middle|\; (x,y)_{1:t-1}\right] - \inf_{f\in\mathcal{F}} \sum_{t=1}^{T} |y_t - f(x_t)|\right].$$

To this end, choose any $\mathcal{X}$-valued tree $\mathbf{x}$ of depth T. Let $y_1, \ldots, y_T$ be i.i.d. Rademacher random variables and define $x_t = \mathbf{x}(y_{1:t-1})$ deterministically (that is, the conditional distribution of x_t is a point distribution on $\mathbf{x}(y_{1:t-1})$). It is easy to see that this distribution makes the choice f_t irrelevant, yielding

$$\mathcal{V}_T^{\mathrm{S}}(\mathcal{F}) \ge \mathbb{E}\left[\sum_{t=1}^{T} 1 - \inf_{f\in\mathcal{F}} \sum_{t=1}^{T} |y_t - f(x_t)|\right] = \mathbb{E}_{y_1,\ldots,y_T} \sup_{f\in\mathcal{F}} \sum_{t=1}^{T} y_t f(x_t).$$

Since this holds for any tree $\mathbf{x}$, we obtain the desired lower bound $\mathcal{V}_T^{\mathrm{S}}(\mathcal{F}) \ge \mathfrak{R}_T(\mathcal{F})$. The final lower bound on $\mathfrak{R}_T(\mathcal{F})$ (in terms of the fat-shattering dimensions) is proved by Rakhlin et al. (2014, Lemma 2).

■

Proof [**of Theorem 10**] The equivalence of *1* and *2* follows directly from Proposition 9. First, suppose that fat_α is infinite for some $\alpha > 0$. Then, the lower bound says that $\mathcal{V}_T^{\mathrm{S}}(\mathcal{F}) \ge \alpha T/(4\sqrt{2})$ and hence $\limsup_{T\to\infty} \mathcal{V}_T^{\mathrm{S}}(\mathcal{F})/T \ge \alpha/(4\sqrt{2})$. Thus, the class $\mathcal{F}$ is not online learnable in the supervised setting. Now, assume that fat_α is finite for all α. Fix an $\epsilon > 0$ and choose $\alpha = \epsilon/16$. Using the upper bound, we have

$$\begin{aligned}
\mathcal{V}_T^{\mathrm{S}}(\mathcal{F}) &\le 8T\alpha + 24\sqrt{T}\int_\alpha^1 \sqrt{\mathrm{fat}_\beta \log\left(\frac{2eT}{\beta}\right)}\,d\beta \\
&\le 8T\alpha + 24\sqrt{T}(1-\alpha)\sqrt{\mathrm{fat}_\alpha \log\left(\frac{2eT}{\alpha}\right)} \\
&\le \epsilon T/2 + \epsilon T/2
\end{aligned}$$

for T large enough. Thus, $\limsup_{T\to\infty} \mathcal{V}_T^{\mathrm{S}}(\mathcal{F})/T \le \epsilon$. Since $\epsilon > 0$ was arbitrary, this proves that $\mathcal{F}$ is online learnable in the supervised setting.

The statement that $\mathcal{V}_T^{\mathrm{S}}(\mathcal{F})$, $\mathfrak{R}_T(\mathcal{F})$, and $\mathfrak{D}_T(\mathcal{F})$ are within a multiplicative factor of $\mathcal{O}(\log^{3/2} T)$ of each other whenever the problem is online learnable follows immediately from Eq. (10) in (Rakhlin et al., 2014) and Proposition 9.

■

Proof [**of Lemma 13**] Consider the game $(\mathcal{F}, \mathcal{Z}_{\mathrm{cvx}})$ and fix a randomized strategy π of the player. Then, the expected regret of a randomized strategy π against any adversary playing $g_1, \ldots, g_T$ can be lower-bounded via Jensen's inequality as

$$\sum_{t=1}^{T} \mathbb{E}_{u_t \sim \pi_t(g_{1:t-1})}\left[g_t(u_t)\right] - \inf_{u\in\mathcal{F}} \sum_{t=1}^{T} g_t(u) \ge \sum_{t=1}^{T} g_t\left(\mathbb{E}_{u_t \sim \pi_t(g_{1:t-1})}\left[u_t\right]\right) - \inf_{u\in\mathcal{F}} \sum_{t=1}^{T} g_t(u),$$

which is simply regret of a *deterministic* strategy obtained from π by playing $\mathbb{E}_{u_t \sim \pi_t(g_{1:t-1})}\left[u_t\right]$ on round t. Thus, to any randomized strategy corresponds a deterministic one that is no worse. On the other hand, the set of randomized strategies contains the set of deterministic ones. Hence, $\mathcal{V}_T(\mathcal{F}, \mathcal{Z}_{\mathrm{cvx}}) = \mathcal{V}_T^{\mathrm{det}}(\mathcal{F}, \mathcal{Z}_{\mathrm{cvx}})$ where $\mathcal{V}_T^{\mathrm{det}}$ is defined as the minimax regret obtainable only using deterministic player strategies. Now, we appeal to Theorem 14 of Abernethy et al. (2008) that says $\mathcal{V}_T^{\mathrm{det}}(\mathcal{F}, \mathcal{Z}_{\mathrm{cvx}}) = \mathcal{V}_T^{\mathrm{det}}(\mathcal{F}, \mathcal{Z}_{\mathrm{lin}})$. Note that Abernethy et al. (2008) deal with convex sets in finite dimensional spaces only. However, their proof relies on fundamental properties of convex functions that are true in any general vector space (such as the fact that the first order Taylor expansion of a convex function globally lower bounds the convex function). Since $\mathcal{Z}_{\mathrm{lin}}$ also consists of convex (in fact, linear) functions, the above argument again gives $\mathcal{V}_T^{\mathrm{det}}(\mathcal{F}, \mathcal{Z}_{\mathrm{lin}}) = \mathcal{V}_T(\mathcal{F}, \mathcal{Z}_{\mathrm{lin}})$. This finishes the proof of the lemma. ■

Proof [**of Proposition 15**] We shall prove that for any $i \in \{2, \ldots, k\}$,

$$\mathfrak{R}_T(\mathcal{F}_i) \le 16 L B_i \left(1 + 4\sqrt{2}\log^{3/2}(eT^2)\right) \mathfrak{R}_T(\mathcal{F}_{i-1}).$$

To see this note that for any $\mathbf{x}$, $\mathfrak{R}_T(\mathcal{F}_i, \mathbf{x})$ is equal to

$$\mathbb{E}_\epsilon\left[\sup_{\substack{w^i:\|w^i\|_1\le B_i \\ \forall j\ f_j\in\mathcal{F}_{i-1}}} \sum_{t=1}^{T} \epsilon_t \left(\sum_j w_j^i \sigma\left(f_j(\mathbf{x}_t(\epsilon))\right)\right)\right] \le \mathbb{E}_\epsilon\left[\sup_{\substack{w^i:\|w^i\|_1\le B_i \\ \forall j\ f_j\in\mathcal{F}_{i-1}}} \|w^i\|_1 \max_j \left|\sum_{t=1}^{T} \epsilon_t \sigma\left(f_j(\mathbf{x}_t(\epsilon))\right)\right|\right]$$

by Hölder's inequality. Then $\mathfrak{R}_T(\mathcal{F}_i)$ is upper bounded as

$$\begin{aligned}
&\sup_{\mathbf{x}} \mathbb{E}_\epsilon\left[B_i \sup_{f\in\mathcal{F}_{i-1}} \max\left\{\sum_{t=1}^{T} \epsilon_t \sigma\left(f(\mathbf{x}_t(\epsilon))\right), -\sum_{t=1}^{T} \epsilon_t \sigma\left(f(\mathbf{x}_t(\epsilon))\right)\right\}\right] \\
&\le \sup_{\mathbf{x}} \mathbb{E}_\epsilon\left[B_i \max\left\{\sup_{f\in\mathcal{F}_{i-1}} \sum_{t=1}^{T} \epsilon_t \sigma\left(f(\mathbf{x}_t(\epsilon))\right), \sup_{f\in\mathcal{F}_{i-1}} \sum_{t=1}^{T} -\epsilon_t \sigma\left(f(\mathbf{x}_t(\epsilon))\right)\right\}\right].
\end{aligned}$$

Since $0 \in \mathcal{F}_i$ together with the assumption of $\sigma(0) = 0$, both terms are non-negative, and thus the maximum above can be upper bounded by the sum

$$\sup_{\mathbf{x}} \mathbb{E}_\epsilon \left[B_i \sup_{f \in \mathcal{F}_{i-1}} \sum_{t=1}^{T} \epsilon_t \sigma\left(f(\mathbf{x}_t(\epsilon))\right) \right] + \sup_{\mathbf{x}} \mathbb{E}_\epsilon \left[B_i \sup_{f \in \mathcal{F}_{i-1}} \sum_{t=1}^{T} -\epsilon_t \sigma\left(f(\mathbf{x}_t(\epsilon))\right) \right] .$$

We now claim that the two terms are equal. Indeed, let $\mathbf{x}^*$ be the tree achieving the supremum in the first term (a modified analysis can be carried out if the supremum is not achieved). Then the mirror tree $\mathbf{x}$ defined via $\mathbf{x}_t(\epsilon) = \mathbf{x}_t^*(-\epsilon)$ yields the same value for the second term. Since the argument can be carried out in the reverse direction, the two terms are equal, and the upper bound of

$$2B_i \sup_{\mathbf{x}} \mathbb{E}_\epsilon \left[\sup_{f \in \mathcal{F}_{i-1}} \sum_{t=1}^{T} \epsilon_t \sigma\left(f(\mathbf{x}_t(\epsilon))\right) \right]$$

follows. In view of contraction in Corollary 5, we obtain a further upper bound of

$$16 B_i L \left(1 + 4\sqrt{2} \log^{3/2}(eT^2)\right) \mathfrak{R}_T(\mathcal{F}_{i-1}). \tag{29}$$

To finish the proof we note that for the base case of $i = 1$, $\mathfrak{R}_T(\mathcal{F}_1)$ is equal to

$$\sup_{\mathbf{x}} \mathbb{E}_\epsilon \left[\sup_{w \in \mathbb{R}^d : \|w\|_1 \le B_1} \sum_{t=1}^{T} \epsilon_t w^\top \mathbf{x}_t(\epsilon) \right]$$

which is upper bounded by

$$\sup_{\mathbf{x}} \mathbb{E}_\epsilon \left[\sup_{w \in \mathbb{R}^d : \|w\|_1 \le B_1} \|w\|_1 \left\| \sum_{t=1}^{T} \epsilon_t \mathbf{x}_t(\epsilon) \right\|_\infty \right] \le B_1 \sup_{\mathbf{x}} \mathbb{E}_\epsilon \left[\max_{i \in [d]} \left\{ \sum_{t=1}^{T} \epsilon_t \mathbf{x}_t(\epsilon)[i] \right\} \right].$$

Note that the instances $x \in \mathcal{X}$ are vectors in $\mathbb{R}^d$ and so for a given instance tree $\mathbf{x}$, for any $i \in [d]$, $\mathbf{x}[i]$ given by only taking the i^{th} co-ordinate is a valid real valued tree. By (4),

$$T \cdot \mathfrak{R}_T(\mathcal{F}_1) \le B_1 \sup_{\mathbf{x}} \mathbb{E}_\epsilon \left[\max_{i \in [d]} \left\{ \sum_{t=1}^{T} \epsilon_t \mathbf{x}_t(\epsilon)[i] \right\} \right] \le B_1 \sqrt{2TX_\infty^2 \log d}.$$

Using the above and (29) repeatedly we conclude the proof. ■

Proof [**of Proposition 16**] Fix a $\gamma > 0$ and use loss

$$\ell(\hat{y}, y) = \begin{cases} 1 & \hat{y}y \le 0 \\ 1 - \hat{y}y/\gamma & 0 < \hat{y}y < \gamma \\ 0 & \hat{y}y \ge \gamma \end{cases}$$

Since this loss is $1/\gamma$-Lipschitz, we can use (11) and the Rademacher contraction Corollary 5 to show that for each $\gamma > 0$ there exists a randomized strategy τ^γ such that for any data sequence

$$\sum_{t=1}^{T} \mathbb{E}_{\hat{y}_t \sim \tau_t^\gamma(z_{1:t-1})} \left[\ell(\hat{y}_t, y_t) \right] \le \inf_{f \in \mathcal{F}} \sum_{t=1}^{T} \ell(f(x_t), y_t) + \gamma^{-1} \rho_T T \mathfrak{R}_T(\mathcal{F}),$$

where $\rho_T = 16\left(1 + 4\sqrt{2}\log^{3/2}(eT^2)\right)$ throughout the proof. Further, observe that the loss function is lower bounded by the zero-one loss $\mathbf{1}\{\hat{y}y < 0\}$ and is upper bounded by the margin zero-one loss $\mathbf{1}\{\hat{y}y < \gamma\}$. Hence,

$$\sum_{t=1}^{T} \mathbb{E}_{\hat{y}_t \sim \tau_t^\gamma(z_{1:t-1})}\left[\mathbf{1}\{\hat{y}_t y_t < 0\}\right] \le \inf_{f\in\mathcal{F}} \sum_{t=1}^{T} \mathbf{1}\{y_t f(x_t) < \gamma\} + \gamma^{-1}\rho_T T \mathfrak{R}_T(\mathcal{F}). \tag{30}$$

The above bound holds for randomized each strategy given by τ^γ, for any given γ. Now we discretize the set of γ's as $\gamma_i = 1/2^i$ and use the output of the randomized strategies $\tau^{\gamma_1}, \tau^{\gamma_2}, \ldots$, that attain the regret bounds given in (30), as experts. We then run a countable experts algorithm (Algorithm 1) with initial weight for expert i as $p_i = \frac{6}{\pi^2 i^2}$. Such an algorithm achieves $\mathcal{O}(\sqrt{T}\log(1/p_i))$ regret w.r.t. expert i. In view of Proposition 20, for this randomized strategy τ, for any i,

$$\sum_{t=1}^{T} \mathbb{E}_{\hat{y}_t \sim \tau_t(z_{1:t-1})}\left[\mathbf{1}\{\hat{y}_t y_t < 0\}\right] \le \inf_{f\in\mathcal{F}} \sum_{t=1}^{T} \mathbf{1}\{y_t f(x_t) < \gamma_i\} + \gamma_i^{-1}\rho_T T \mathfrak{R}_T(\mathcal{F}) + \sqrt{T}\left(1 + 2\log\left(\frac{i\pi}{\sqrt{6}}\right)\right).$$

For any $\gamma > 0$, let $i_\gamma \in 0, 1, \ldots,$ be such that $2^{-(i_\gamma+1)} < \gamma \le 2^{-i_\gamma}$. Then above right-hand side is upper bounded by

$$\inf_{f\in\mathcal{F}} \sum_{t=1}^{T} \mathbf{1}\{y_t f(x_t) < 2\gamma\} + \gamma^{-1}\rho_T T \mathfrak{R}_T(\mathcal{F}) + \sqrt{T}\left(1 + 2\log\left(\frac{i_\gamma\pi}{\sqrt{6}}\right)\right).$$

The proof is concluded using the inequality $i_\gamma \le \log(1/\gamma)$ and upper bounding constants. ■

Proof [**of Proposition 17**] Fix some $L > 0$. The loss

$$\phi_L(\alpha) = \begin{cases} 1 & \text{if } \alpha \le 0 \\ 1 - L\alpha & \text{if } 0 < \alpha \le 1/L \\ 0 & \text{otherwise} \end{cases}$$

is L-Lipschitz and so by Theorem 7 and Corollary 5 we have that for every $L > 0$, there exists a randomized strategy τ^L for the player, such that for any sequence $z_1 = (x_1, y_1), \ldots, z_T = (x_T, y_T)$,

$$\sum_{t=1}^{T} \mathbb{E}_{\hat{y}_t \sim \tau_t^L(z_{1:t-1})}\left[\phi_L(y_t\hat{y}_t)\right] \le \inf_{f\in\mathcal{F}} \sum_{t=1}^{T} \phi_L(y_t f(x_t)) + L\rho_T T \mathfrak{R}_T(\mathcal{F}), \tag{31}$$

where $\rho_T = 16\left(1 + 4\sqrt{2}\log^{3/2}(eT^2)\right)$ throughout this proof. Since ϕ_L dominates the step function, the left hand side of (31) also upper-bounds the expected indicator loss

$$\sum_{t=1}^{T} \mathbb{E}_{\hat{y}_t \sim \tau_t^L(z_{1:t-1})}\left[\mathbf{1}\{\hat{y}_t \ne y_t\}\right].$$

For any $f \in \mathcal{F}$, we can relate the ϕ_L-loss to the indicator loss by

$$\sum_{t=1}^{T} \phi_L(y_t f(x_t)) = \sum_{t=1}^{T} \mathbf{1}\{y_t f(x_t) \le 0\} + \sum_{l} C(l)\phi_L(w_l).$$

Let us now use the above decomposition in (31). Crucially, the sign of $f(x)$ does not depend on w_l, but only on the label σ_l of the unique leaf l reached by x. Thus, the infimum in (31) can be split into two infima:

$$\inf_{f\in\mathcal{F}}\sum_{t=1}^{T}\phi_L(y_t f(x_t)) = \inf_{f\in\mathcal{F}}\sum_{t=1}^{T}\mathbf{1}\left\{y_t f(x_t)\le 0\right\} + \inf_{w_l}\sum_{l} C(l)\phi_L(w_l),$$

where it is understood that the $C(l)$ term on the right hand side is computed using the function f minimizing the first sum on the right hand side. We can further write

$$\sum_l C(l)\phi_L(w_l) \le \sum_l C(l)\max(0, 1 - Lw_l) = \sum_l \max\left(0, (1-Lw_l)C(l)\right).$$

So far, we have derived a regret bound for a given L. Let us now remove the requirement to know L a priori by running the experts Algorithm 1 with $\tau^1, \tau^2, \ldots$ as a countable set of experts corresponding to the values $L \in \mathbb{N}$. The prior on expert L is taken to be $p_L = \frac{6}{\pi^2}L^{-2}$ so that $\sum p_L = 1$. For the randomized strategy τ obtained in this manner, from Proposition 20, for any sequence of instances and any $L \in \mathbb{N}$,

$$\begin{aligned}\sum_{t=1}^{T}\mathbb{E}_{\hat{y}_t\sim\tau_t(z_{1:t-1})}\left[\mathbf{1}\left\{\hat{y}\neq y_t\right\}\right] \le \inf_{f\in\mathcal{F}}\sum_{t=1}^{T}\mathbf{1}\left\{y_t f(x_t)\le 0\right\} + \inf_{f\in\mathcal{F}}\sum_l \max\left(0,(1-Lw_l)C(l)\right)\\ + L\rho_T T\mathfrak{R}_T(\mathcal{F}) + \sqrt{T} + 2\sqrt{T}\log(L\pi/\sqrt{6}).\end{aligned}$$

Now we pick $L = |\{l : C(l) > \rho_T T\mathfrak{R}_T(\mathcal{F})\}| \le N$ and upper bound the second infimum by choosing $w_l = 0$ if $C(l) \le \rho_T T\mathfrak{R}_T(\mathcal{F})$ and $w_l = 1/L$ otherwise:

$$\begin{aligned}\inf_{w_l}\sum_l \max\left(0,(1-Lw_l)C(l)\right) + L\rho_T T\mathfrak{R}_T(\mathcal{F}) \le \sum_l C(l)\mathbf{1}\left\{C(l)\le\rho_T T\mathfrak{R}_T(\mathcal{F})\right\}\\ + \rho_T T\mathfrak{R}_T(\mathcal{F})\sum_l \mathbf{1}\left\{C(l) > \rho_T T\mathfrak{R}_T(\mathcal{F})\right\}\end{aligned}$$

which can be written succinctly as

$$\sum_l \min\{C(l), \rho_T T\mathfrak{R}_T(\mathcal{F})\}.$$

We conclude that

$$\begin{aligned}\sum_{t=1}^{T}\mathbb{E}_{\hat{y}_t\sim\tau_t(z_{1:t-1})}\left[\mathbf{1}\left\{\hat{y}_t\neq y_t\right\}\right] \le \inf_{f\in\mathcal{F}}\sum_{t=1}^{T}\mathbf{1}\left\{y_t f(x_t)\le 0\right\}\\ + \sum_l \min(C(l), \rho_T T\mathfrak{R}_T(\mathcal{F})) + \sqrt{T}\left(1 + 2\log(N\pi/\sqrt{6})\right).\end{aligned}$$

Finally, we apply Corollary 6 and Lemma 3(2) to bound $\mathfrak{R}_T(\mathcal{F}) \le d\mathcal{O}(\log^{3/2} T)\ \mathfrak{R}_T(\mathcal{H})$ and thus conclude the proof. ■

Proof [**of Proposition 18**] First, by the classical result of Kolmogorov and Tikhomirov (1959), the class $\mathcal{G}$ of all bounded Lipschitz functions on a bounded interval has small metric

entropy: $\log \widehat{\mathcal{N}}_\infty(\alpha, \mathcal{G}) = \Theta(1/\alpha)$. For the particular class of non-decreasing 1-Lipschitz functions, it is trivial to verify that the entropy is in fact bounded by $2/\alpha$. Considering all 1-Lipschitz functions increases this to c_0/α for some universal constant c_0.

Next, consider the class $\mathcal{F} = \{\langle w, x\rangle \mid \|w\|_2 \le 1\}$ over the Euclidean ball. By Proposition 14, $\mathfrak{R}_T(\mathcal{F}) \le 1/\sqrt{T}$. Using the lower bound of Proposition 9, $\mathrm{fat}_\alpha \le 32/\alpha^2$ whenever $\alpha > 4\sqrt{2}/\sqrt{T}$. This implies that $\mathcal{N}_\infty(\alpha, \mathcal{F}, T) \le (2eT/\alpha)^{32/\alpha^2}$ whenever $\alpha > 4\sqrt{2}/\sqrt{T}$. Note that this bound does not depend on the ambient dimension of $\mathcal{X}$.

Next, we show that a composition of $\mathcal{G}$ with any "small" class $\mathcal{F} \subset [-1,1]^{\mathcal{X}}$ also has a small cover. To this end, suppose $\mathcal{N}_\infty(\alpha, \mathcal{F}, T)$ is the covering number for $\mathcal{F}$. Fix a particular tree $\mathbf{x}$ and let $V = \{\mathbf{v}_1, \ldots, \mathbf{v}_N\}$ be an ℓ_∞ cover of $\mathcal{F}$ on $\mathbf{x}$ at scale α. Analogously, let $W = \{g_1, \ldots, g_M\}$ be an ℓ_∞ cover of $\mathcal{G}$ with $M = \widehat{\mathcal{N}}_\infty(\alpha, \mathcal{G})$. Consider the class $\mathcal{G} \circ \mathcal{F} = \{g \circ f : g \in \mathcal{G}, f \in \mathcal{F}\}$. The claim is that $\{g(\mathbf{v}) : \mathbf{v} \in V, g \in W\}$ provides an ℓ_∞ cover for $\mathcal{G} \circ \mathcal{F}$ on $\mathbf{x}$. Fix any $f \in \mathcal{F}, g \in \mathcal{G}$ and $\epsilon \in \{\pm 1\}^T$. Let $\mathbf{v} \in V$ be such that $\max_{t \in [T]} |f(\mathbf{x}_t(\epsilon)) - \mathbf{v}_t(\epsilon)| \le \alpha$, and let $g' \in W$ be such that $\|g - g'\|_\infty \le \alpha$. Then, using the fact that functions in $\mathcal{G}$ are 1-Lipschitz, for any $t \in [T]$,

$$|g(f(\mathbf{x}_t(\epsilon))) - g'(\mathbf{v}_t(\epsilon))| \le |g(f(\mathbf{x}_t(\epsilon))) - g'(f(\mathbf{x}_t(\epsilon))| + |g'(f(\mathbf{x}_t(\epsilon)) - g'(\mathbf{v}_t(\epsilon))| \le 2\alpha \ .$$

Hence, $\mathcal{N}_\infty(2\alpha, \mathcal{G} \circ \mathcal{F}, T) \le \widehat{\mathcal{N}}_\infty(\alpha, \mathcal{G}) \times \mathcal{N}_\infty(\alpha, \mathcal{F}, T)$.

Finally, we put all the pieces together. By Theorem 8, the minimax value is bounded by $8T$ times the sequential Rademacher complexity of the class $\mathcal{G} \circ \mathcal{F} = \{u(\langle w, x\rangle) \mid u : [-1,1] \to [-1,1] \text{ is 1-Lipschitz }, \|w\|_2 \le 1\}$ since the squared loss is 4-Lipschitz on the space of possible values. The latter complexity is then bounded by

$$\begin{aligned} T\mathfrak{D}_T(\mathcal{G} \circ \mathcal{F}) &\le 32\sqrt{T} + 12 \int_{8/\sqrt{T}}^1 \sqrt{T \ \log \ \mathcal{N}(\delta, \mathcal{G} \circ \mathcal{F}, T)} \, d\delta \\ &\le 32\sqrt{T} + 12\sqrt{T} \int_{8/\sqrt{T}}^1 \sqrt{\frac{4c_0}{\delta} + \frac{128}{\delta^2} \log(2eT)} d\delta \ . \end{aligned}$$

We therefore conclude that the value of the game for the supervised learning problem is bounded by $\mathcal{O}(\sqrt{T} \log^{3/2}(T))$. ■

Appendix C. Exponentially Weighted Average (EWA) Algorithm on Countable Experts

We consider here a version of the exponentially weighted experts algorithm for a countable (possibly infinite) number of experts and provide a bound on the expected regret of the randomized algorithm. The proof of the result closely follows the finite case (e.g., Cesa-Bianchi and Lugosi, 2006, Theorem 2.2). This result is well known and we include it here for completeness, as it is needed in the proofs of Proposition 16 and Proposition 17.

Suppose we are provided with countable experts $E_1, E_2, \ldots$, where each expert can herself be thought of as a randomized/deterministic player strategy which, given history, produces an element of $\mathcal{F}$ at round t. Here we also assume that $\mathcal{F} \subseteq [0,1]^{\mathcal{X}}$. Denote by f_t^i the function output by expert i at round t given the history. The EWA algorithm we consider needs access to the countable set of experts and also needs an initial weighting on each expert $p_1, p_2, \ldots$ such that $\sum_i p_i = 1$.

Algorithm 1 EWA $(E_1, E_2, \ldots, p_1, p_2, \ldots)$

Initialize each $w_i^1 \leftarrow p_i$
for $t = 1$ to T **do**
  Pick randomly an expert i with probability w_i^t
  Play $f_t = f_i^t$
  Receive x_t
  Update for each i, $w_i^{t+1} = \frac{w_i^t e^{-\eta f_i^t(x_t)}}{\sum_i w_i^t e^{-\eta f_i^t(x_t)}}$
end for

Proposition 20 *The exponentially weighted average forecaster (Algorithm 1) with* $\eta = T^{-1/2}$ *enjoys the regret bound*

$$\sum_{t=1}^{T} \mathbb{E}\left[f_t(x_t)\right] \le \sum_{t=1}^{T} f_i^t(x_t) + \frac{\sqrt{T}}{8} + \sqrt{T}\log\left(1/p_i\right)$$

for any $i \in \mathbb{N}$.

References

J. Abernethy, P. L. Bartlett, A. Rakhlin, and A. Tewari. Optimal strategies and minimax lower bounds for online convex games. In *Proceedings of the 21st Annual Conference on Learning Theory*, pages 414–424. Omnipress, 2008.

J. Abernethy, A. Agarwal, P. L. Bartlett, and A. Rakhlin. A stochastic view of optimal regret through minimax duality. In *Proceedings of the 22nd Annual Conference on Learning Theory*, 2009.

P. L. Bartlett and S. Mendelson. Rademacher and Gaussian complexities: risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2003.

S. Ben-David, D. Pal, and S. Shalev-Shwartz. Agnostic online learning. In *Proceedings of the 22th Annual Conference on Learning Theory*, 2009.

D. Blackwell. An analog of the minimax theorem for vector payoffs. *Pacific Journal of Mathematics*, 6(1):1–8, 1956a.

D. Blackwell. Controlled random walks. In *Proceedings of the International Congress of Mathematicians, 1954*, volume 3, pages 336–338. North Holland, 1956b.

V.I. Bogachev. *Measure Theory*, volume 2. Springer, 2007. ISBN 3540345132.

J.M. Borwein. A very complicated proof of the minimax theorem. *Minimax Theory and Its Applications*, 1(1), 2014.

J.M. Borwein and D Zhuang. On Fan's minimax theorem. *Mathematical programming*, 34 (2):232–234, 1986.

N. Cesa-Bianchi and G. Lugosi. On prediction of individual sequences. *Annals of Statistics*, pages 1865–1895, 1999.

N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

N. Cesa-Bianchi, Y. Freund, D. Haussler, D. P. Helmbold, R. E. Schapire, and M. K. Warmuth. How to use expert advice. *Journal of the ACM*, 44(3):427–485, 1997.

T. Cover. Behavior of sequential predictors of binary sequences. In *Transactions of the Fourth Prague Conference on Information Theory, Statistical Decision Functions, Random Processes, 1965*, pages 263–272. Publishing House of the Czechoslovak Academy of Sciences, 1967.

T. M. Cover and A. Shenhar. Compound Bayes predictors for sequences with apparent Markov structure. *IEEE Transactions on Systems, Man and Cybernetics*, 7(6):421–424, 1977.

L. Davisson. Universal noiseless coding. *Information Theory, IEEE Transactions on*, 19 (6):783–795, 1973.

M. Feder, N. Merhav, and M. Gutman. Universal prediction of individual sequences. *Information Theory, IEEE Transactions on*, 38(4):1258–1270, 1992.

D. P. Foster and R. V. Vohra. Calibrated learning and correlated equilibrium. *Games and Economic Behavior*, 21(1):40–55, 1997.

J. Hannan. Approximation to Bayes risk in repeated play. *Contributions to the Theory of Games*, 3:97–139, 1957.

S. Hart and A. Mas-Colell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68(5):1127–1150, 2000.

S. M. Kakade and A. T. Kalai. From batch to transductive online learning. In Y. Weiss, B. Schölkopf, and J.C. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 611–618. MIT Press, 2006.

A. Kalai and S. Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.

A. T. Kalai and R. Sastry. The isotron algorithm: High-dimensional isotonic regression. In *Proceedings of the 22th Annual Conference on Learning Theory*, 2009.

A.N. Kolmogorov and V.M. Tikhomirov. ε-entropy and ε-capacity of sets in function spaces. *Uspekhi Matematicheskikh Nauk*, 14(2):3–86, 1959.

V. Koltchinskii and D. Panchenko. Empirical margin distributions and bounding the generalization error of combined classifiers. *Annals of Statistics*, 30(1):1–50, 2002.

M. Ledoux and M. Talagrand. *Probability in Banach Spaces*. Springer-Verlag, New York, 1991.

N. Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2(4):285–318, 04 1988.

N. Littlestone and M. K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994.

A. Rakhlin and K. Sridharan. Statistical learning and sequential prediction, 2014. Available at `http://stat.wharton.upenn.edu/~rakhlin/courses/stat928/stat928_notes.pdf`.

A. Rakhlin, K. Sridharan, and A. Tewari. Online learning: Random averages, combinatorial parameters, and learnability. In *Advances in Neural Information Processing Systems 23*, pages 1984–1992, 2010.

A. Rakhlin, K. Sridharan, and A. Tewari. Online learning: Beyond regret. In *Proceedings of the 24th Annual Conference on Learning Theory*, volume 19 of *JMLR Workshop and Conference Proceedings*, pages 559–594, 2011.

A. Rakhlin, O. Shamir, and K. Sridharan. Relax and randomize: From value to algorithms. In *Advances in Neural Information Processing Systems 25*, pages 2150–2158, 2012.

A. Rakhlin, K. Sridharan, and A. Tewari. Sequential complexities and uniform laws of large numbers. *Probability Theory and Related Fields*, 2014.

J. Rissanen. Universal coding, information, prediction, and estimation. *Information Theory, IEEE Transactions on*, 30(4):629–636, 1984.

H. Robbins. Asymptotically subminimax solutions of compound statistical decision problems. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 131–149. University of California Press, 1950.

R. E. Schapire, Y. Freund, P. Bartlett, and W.S. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, pages 322–330, 1997.

S. Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2011.

S. Shalev-Shwartz, O. Shamir, N. Srebro, and K. Sridharan. Stochastic convex optimization. In *Conference on Learning Theory*, 2009.

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

V. Vovk. A game of prediction with expert advice. *Journal of Computer and System Sciences*, 56(2):153–173, 1998.

M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 928–936, 2003.

J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *Information Theory, IEEE Transactions on*, 23(3):337–343, 1977.

Journal of Machine Learning Research 16 (2015) 187-225 Submitted 9/13; Revised 6/14; Published 2/15

Learning Transformations for Clustering and Classification

Qiang Qiu QIANG.QIU@DUKE.EDU
Department of Electrical and Computer Engineering
Duke University
Durham, NC 27708, USA

Guillermo Sapiro GUILLERMO.SAPIRO@DUKE.EDU
Department of Electrical and Computer Engineering
Department of Computer Science
Department of Biomedical Engineering
Duke University
Durham, NC 27708, USA

Editor: Ben Recht

Abstract

A low-rank transformation learning framework for subspace clustering and classification is proposed here. Many high-dimensional data, such as face images and motion sequences, approximately lie in a union of low-dimensional subspaces. The corresponding subspace clustering problem has been extensively studied in the literature to partition such high-dimensional data into clusters corresponding to their underlying low-dimensional subspaces. Low-dimensional intrinsic structures are often violated for real-world observations, as they can be corrupted by errors or deviate from ideal models. We propose to address this by learning a linear transformation on subspaces using nuclear norm as the modeling and optimization criteria. The learned linear transformation restores a low-rank structure for data from the same subspace, and, at the same time, forces a maximally separated structure for data from different subspaces. In this way, we reduce variations within the subspaces, and increase separation between the subspaces for a more robust subspace clustering. This proposed learned robust subspace clustering framework significantly enhances the performance of existing subspace clustering methods. Basic theoretical results presented here help to further support the underlying framework. To exploit the low-rank structures of the transformed subspaces, we further introduce a fast subspace clustering technique, which efficiently combines robust PCA with sparse modeling. When class labels are present at the training stage, we show this low-rank transformation framework also significantly enhances classification performance. Extensive experiments using public data sets are presented, showing that the proposed approach significantly outperforms state-of-the-art methods for subspace clustering and classification. The learned low cost transform is also applicable to other classification frameworks.

Keywords: subspace clustering, classification, low-rank transformation, nuclear norm, feature learning

1. Introduction

High-dimensional data often have a small intrinsic dimension. For example, in the area of computer vision, face images of a subject (Basri and Jacobs, 2003; Wright et al., 2009),

handwritten images of a digit (Hastie and Simard, 1998), and trajectories of a moving object (Tomasi and Kanade, 1992) can all be well-approximated by a low-dimensional subspace of the high-dimensional ambient space. Thus, multiple class data often lie in a union of low-dimensional subspaces. The ubiquitous subspace clustering problem is to partition high-dimensional data into clusters corresponding to their underlying subspaces.

Standard clustering methods such as k-means in general are not applicable to subspace clustering. Various methods have been recently suggested for subspace clustering, such as Sparse Subspace Clustering (SSC) (Elhamifar and Vidal, 2013), and its extensions (Liu et al., 2010; Soltanolkotabi and Candes, 2012; Soltanolkotabi et al., 2013; Wang and Xu, 2013), Local Subspace Affinity (LSA) (Yan and Pollefeys, 2006), Local Best-fit Flats (LBF) (Zhang et al., 2012), Generalized Principal Component Analysis (Vidal et al., 2003), Agglomerative Lossy Compression (Ma et al., 2007), Locally Linear Manifold Clustering (Goh and Vidal, 2007), and Spectral Curvature Clustering (Chen and Lerman, 2009). A recent survey on subspace clustering can be found in Vidal (2011).

Low-dimensional intrinsic structures, which enable subspace clustering, are often violated for real-world data. For example, under the assumption of Lambertian reflectance, Basri and Jacobs (2003) show that face images of a subject obtained under a wide variety of lighting conditions can be accurately approximated with a 9-dimensional linear subspace. However, real-world face images are often captured under pose variations; in addition, faces are not perfectly Lambertian, and exhibit cast shadows and specularities (Candès et al., 2011). Therefore, it is critical for subspace clustering to handle corrupted underlying structures of realistic data, and as such, deviations from ideal subspaces.

When data from the same low-dimensional subspace are arranged as columns of a single matrix, the matrix should be approximately low-rank. Thus, a promising way to handle corrupted data for subspace clustering is to restore such low-rank structure. Recent efforts have been invested in seeking transformations such that the transformed data can be decomposed as the sum of a low-rank matrix component and a sparse error one (Peng et al., 2010; Shen and Wu, 2012; Zhang et al., 2011). Peng et al. (2010) and Zhang et al. (2011) are proposed for image alignment, Kuybeda et al. (2013) for the extension to multiple-classes with applications in cryo-tomograhy, and Shen and Wu (2012) is discussed in the context of salient object detection. All these methods build on recent theoretical and computational advances in rank minimization.

In this paper, we propose to improve subspace clustering and classification by learning a linear transformation on subspaces using matrix rank, via its nuclear norm convex surrogate, as the optimization criteria. The learned linear transformation recovers a low-rank structure for data from the same subspace, and, at the same time, forces a maximally separated structure for data from different subspaces (actually high nuclear norm, which as discussed later, improves the separation between the subspaces). In this way, we reduce variations within the subspaces, and increase separations between the subspaces for more accurate subspace clustering and classification.

For example, as shown in Figure 1, after faces are detected and aligned, e.g., using Zhu and Ramanan (2012), our approach learns linear transformations for face images to restore for the same subject a low-dimensional structure. By comparing the last row to the first row in Figure 1, we can easily notice that faces from the same subject across different poses

are more visually similar in the new transformed space, enabling better face clustering and classification across pose.

This paper makes the following main contributions:

- Subspace low-rank transformation (LRT) is introduced and analyzed in the context of subspace clustering and classification;
- A Learned Robust Subspace Clustering framework (LRSC) is proposed to enhance existing subspace clustering methods;
- A discriminative low-rank (nuclear norm) transformation approach is proposed to reduce the variation within the classes and increase separations between the classes for improved classification;
- We propose a specific fast subspace clustering technique, called Robust Sparse Subspace Clustering (R-SSC), by exploiting low-rank structures of the learned transformed subspaces;
- We discuss online learning of subspace low-rank transformation for big data;
- We demonstrate through extensive experiments that the proposed approach significantly outperforms state-of-the-art methods for subspace clustering and classification.

The proposed approach can be considered as a way of learning data features, with such features learned in order to reduce within-class rank (nuclear norm), increase between class separation, and encourage robust subspace clustering. As such, the framework and criteria introduced here can be incorporated into other data classification and clustering problems.

In Section 2, we formulate and analyze the low-rank transformation learning problem. In Sections 3 and 4, we discuss the low-rank transformation for subspace clustering and classification respectively. Experimental evaluations are given in Section 5 on public data sets commonly used for subspace clustering evaluation. Finally, Section 6 concludes the paper.

2. Learning Low-rank Transformations (LRT)

Let $\{\mathcal{S}_c\}_{c=1}^{C}$ be C m-dimensional subspaces of $\mathbb{R}^d$ (not all subspaces are necessarily of the same dimension, this is only assumed here to simplify notation). A data set is denoted as $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^{N} \subseteq \mathbb{R}^d$, with each data point $\mathbf{y}_i$ in one of the C subspaces and arranged as a column of $\mathbf{Y}$. $\mathbf{Y}_c$ denotes the set of points in the c-th subspace $\mathcal{S}_c$, points arranged as columns of the matrix $\mathbf{Y}_c$.

As data points in $\mathbf{Y}_c$ lie in a low-dimensional subspace, the matrix $\mathbf{Y}_c$ is expected to be *low-rank*, and such low-rank structure is critical for accurate subspace clustering. However, as discussed above, this low-rank structure is often violated for real data.

Our proposed approach is to learn a global linear transformation on subspaces. Such linear transformation restores a low-rank structure for data from the same subspace, and, at the same time, encourages a maximally separated structure for data from different subspaces. In this way, we reduce the variation within the subspaces and introduce separations between the subspaces for more robust subspace clustering or classification.

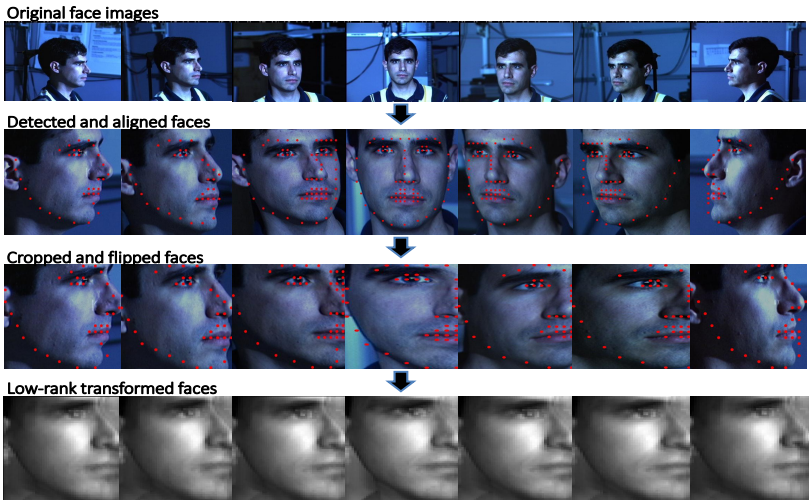


Figure 1: Learned low-rank transformation on faces across pose. In the second row, the input faces are first detected and aligned, e.g., using the method in Zhu and Ramanan (2012). Pose models defined in Zhu and Ramanan (2012) enable an optional crop-and-flip step to retain the more informative side of a face in the third row. Our proposed approach learns linear transformations for face images to restore for the same subject a low-dimensional structure as shown in the last row. By comparing the last row to the first row, we can easily notice that faces from the same subject across different poses are more visually similar in the new transformed space, enabling better face clustering or recognition across pose (note that the goal is clustering/recognition and not reconstruction).

2.1 Preliminary Pedagogical Formulation using Rank

We first assume the data cluster labels are known beforehand, and this assumption is removed when discussing the full clustering approach in Section 3. We adopt matrix rank as the key learning criterion (presented here first for pedagogical reasons, to be later replaced by the nuclear norm), and compute one global linear transformation on all subspaces as

$$\arg\min_{\mathbf{T}} \sum_{c=1}^{C} rank(\mathbf{T}\mathbf{Y}_c) - rank(\mathbf{T}\mathbf{Y}), \quad \text{s.t.} ||\mathbf{T}||_2 = 1, \tag{1}$$

where $\mathbf{T} \in \mathbb{R}^{d \times d}$ is one global linear transformation on all data points (we will later discuss then $\mathbf{T}$'s dimension is less than d), $||\cdot||_2$ denotes the matrix induced 2-norm, and γ is a positive constant. Intuitively, minimizing the first *representation* term $\sum_{c=1}^{C} rank(\mathbf{T}\mathbf{Y}_c)$ encourages a consistent representation for the transformed data from the same subspace; and minimizing the second *discrimination* term $-rank(\mathbf{T}\mathbf{Y})$ encourages a diverse representation for transformed data from different subspaces (we will later formally discuss that the convex surrogate nuclear norm actually has this desired effect). The normalization condition $||\mathbf{T}||_2 = 1$ prevents the trivial solution $\mathbf{T} = 0$.

We now explain that the pedagogical formulation in (1) using rank is however not optimal to simultaneously reduce the variation within the same class subspaces and introduce separations between the different class subspaces, motivating the use of the nuclear norm not only for optimization reasons but for modeling ones as well. Let $\mathbf{A}$ and $\mathbf{B}$ be matrices of the same dimensions (standing for two classes $\mathbf{Y}_1$ and $\mathbf{Y}_2$ respectively), and $[\mathbf{A}, \mathbf{B}]$ (standing for $\mathbf{Y}$) be the concatenation of $\mathbf{A}$ and $\mathbf{B}$, we have (Marsaglia and Styan, 1972)

$$rank([\mathbf{A}, \mathbf{B}]) \leq rank(\mathbf{A}) + rank(\mathbf{B}), \tag{2}$$

with equality if and only if $\mathbf{A}$ and $\mathbf{B}$ are disjoint, i.e., they intersect only at the origin (often the analysis of subspace clustering algorithms considers disjoint spaces, e.g., Elhamifar and Vidal (2013)).

It is easy to show that (2) can be extended for the concatenation of multiple matrices,

$$\begin{aligned} rank([\mathbf{Y}_1, \mathbf{Y}_2, \mathbf{Y}_3, \cdots, \mathbf{Y}_C]) &\leq rank(\mathbf{Y}_1) + rank([\mathbf{Y}_2, \mathbf{Y}_3, \cdots, \mathbf{Y}_C]) \\ &\leq rank(\mathbf{Y}_1) + rank(\mathbf{Y}_2) + rank([\mathbf{Y}_3, \cdots, \mathbf{Y}_C]) \\ &\cdots \\ &\leq \sum_{c=1}^{C} rank(\mathbf{Y}_c), \end{aligned} \tag{3}$$

with equality if matrices are independent. Thus, for (1), we have

$$\sum_{c=1}^{C} rank(\mathbf{T}\mathbf{Y}_c) - rank(\mathbf{T}\mathbf{Y}) \geq 0, \tag{4}$$

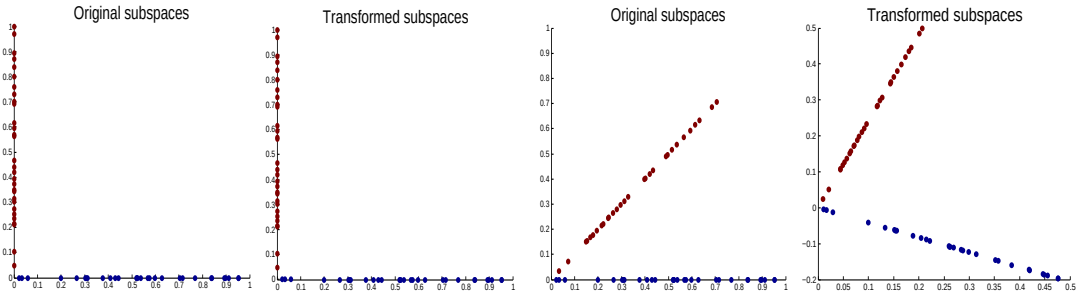
and the objective function (1) reaches the minimum 0 if matrices are independent after applying the learned transformation $\mathbf{T}$. However, independence does not infer maximal separation, an important goal for robust clustering and classification. For example, two lines intersecting only at the origin are independent regardless of the angle in between, and they are maximally separated only when the angle becomes $\frac{\pi}{2}$. With this intuition in mind, we now proceed to describe our proposed formulation based on the nuclear norm.

2.2 Problem Formulation using Nuclear Norm

Let $||\mathbf{A}||_*$ denote the nuclear norm of the matrix $\mathbf{A}$, i.e., the sum of the singular values of $\mathbf{A}$. The nuclear norm $||\mathbf{A}||_*$ is the convex envelop of $rank(\mathbf{A})$ over the unit ball of matrices Fazel (2002). As the nuclear norm can be optimized efficiently, it is often adopted as the best convex approximation of the rank function in the literature on rank optimization, e.g., Candès et al. (2011) and Recht et al. (2010).

One factor that fundamentally affects the performance of subspace clustering and classification algorithms is the distance between subspaces. An important notion to quantify the distance (separation) between two subspaces $\mathcal{S}_i$ and $\mathcal{S}_j$ is the smallest principal angle θ_{ij} (Miao and Ben-Israel, 1992; Elhamifar and Vidal, 2013), which is defined as

$$\theta_{ij} = \min_{\mathbf{u} \in \mathcal{S}_i, \mathbf{v} \in \mathcal{S}_j} \arccos \frac{\mathbf{u}'\mathbf{v}}{||\mathbf{u}||_2 ||\mathbf{v}||_2}, \tag{5}$$

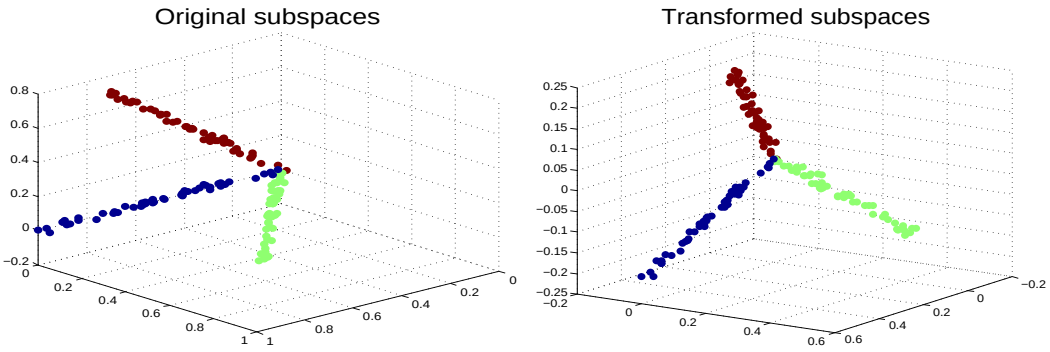


(a) $\theta_{AB} = \frac{\pi}{2} = 1.57$.

(b) $\mathbf{T} = \begin{bmatrix} 1.00 & 0 \\ 0 & 1.00 \end{bmatrix}$; $\theta_{AB} = 1.57$.

(c) $\theta_{AB} = \frac{\pi}{4} = 0.79$, $|\mathbf{A}|_* = 1, |\mathbf{B}|_* = 1$, $|[\mathbf{A}, \mathbf{B}]|_* = 1.41$

(d) $\mathbf{T} = \begin{bmatrix} 0.50 & -0.21 \\ -0.21 & 0.91 \end{bmatrix}$; $\theta_{AB} = 1.57$, $|\mathbf{A}|_* = 1, |\mathbf{B}|_* = 1$, $|[\mathbf{A}, \mathbf{B}]|_* = 1.95$



(e) $\begin{bmatrix} \theta_{AB} = 0.79, & \theta_{AC} = 0.79, & \theta_{BC} = 1.05 \\ \epsilon_A = 0.0141, & \epsilon_B = 0.0131, & \epsilon_C = 0.0148 \end{bmatrix}$, $|\mathbf{A}|_* = 4.06$, $|\mathbf{B}|_* = 4.08$, $|\mathbf{C}|_* = 4.16$.

(f) $\mathbf{T} = \begin{bmatrix} 0.39 & -0.16 & -0.16 \\ -0.13 & 0.90 & 0.11 \\ -0.23 & 0.11 & 0.57 \end{bmatrix}$; $\begin{bmatrix} \theta_{AB} = 1.51, & \theta_{AC} = 1.49, & \theta_{BC} = 1.57 \\ \epsilon_A = 0.0091, & \epsilon_B = 0.0085, & \epsilon_C = 0.0114 \end{bmatrix}$, $|\mathbf{A}|_* = 1.93$, $|\mathbf{B}|_* = 2.37$, $|\mathbf{C}|_* = 1.20$.

Figure 2: The learned transformation $\mathbf{T}$ using (6) with the nuclear norm as the key criterion. Three subspaces in $\mathbb{R}^3$ are denoted as $\mathbf{A}$(red), $\mathbf{B}$(blue), $\mathbf{C}$(green). We denote the angle between subspaces $\mathbf{A}$ and $\mathbf{B}$ as θ_{AB} (and analogous for the other pairs of subspaces). Using (6), we transform $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ in (a),(c),(e) to (b),(d),(f) respectively (in the first row the subspace C is empty, being this basically a two dimensional example). Data points in (e) are associated with random noises $\sim \mathcal{N}(0, 0.01)$. We denote the root mean square deviation of points in $\mathbf{A}$ from the true subspace as ϵ_A (and analogous for the other subspaces). We observe that the learned transformation $\mathbf{T}$ maximizes the distance between every pair of subspaces towards $\frac{\pi}{2}$, and reduces the deviation of points from the true subspace when noise is present, note how the individual subspaces nuclear norm is significantly reduced. Note that, in (c) and (d), we have the same rank values $rank(\mathbf{A}) = 1, rank(\mathbf{B}) = 1, rank([\mathbf{A}, \mathbf{B}]) = 2$, but different nuclear norm values, manifesting the improved between-subspaces separation.

Note that $\theta_{ij} \in [0, \frac{\pi}{2}]$. We replace the rank function in (1) with the nuclear norm,

$$\arg\min_{\mathbf{T}} \sum_{c=1}^{C} ||\mathbf{T}\mathbf{Y}_c||_* - ||\mathbf{T}\mathbf{Y}||_*, \quad \text{s.t.} ||\mathbf{T}||_2 = 1. \tag{6}$$

The normalization condition $||\mathbf{T}||_2 = 1$ prevents the trivial solution $\mathbf{T} = 0$. However, understanding the effects of adopting a different normalization norm here is interesting and is the subject of future research.

It is important to note that (6) is not simply a relaxation of (1). Not only the replacement of the rank by the nuclear norm is critical for optimization considerations in reducing the variation within same class subspaces, but as we show next, the learned transformation $\mathbf{T}$ using the objective function (6) also maximizes the separation between different class subspaces (a missing property in (1)), leading to improved clustering and classification performance.

We start by presenting some basic norm relationships for matrices and their corresponding concatenations.

Theorem 1 *Let* $\mathbf{A}$ *and* $\mathbf{B}$ *be matrices of the same row dimensions, and* $[\mathbf{A}, \mathbf{B}]$ *be the concatenation of* $\mathbf{A}$ *and* $\mathbf{B}$*, we have*

$$||[\mathbf{A}, \mathbf{B}]||_* \leq ||\mathbf{A}||_* + ||\mathbf{B}||_*.$$

Proof: See Appendix A. ■

Theorem 2 *Let* $\mathbf{A}$ *and* $\mathbf{B}$ *be matrices of the same row dimensions, and* $[\mathbf{A}, \mathbf{B}]$ *be the concatenation of* $\mathbf{A}$ *and* $\mathbf{B}$*, we have*

$$||[\mathbf{A}, \mathbf{B}]||_* = ||\mathbf{A}||_* + ||\mathbf{B}||_*.$$

when the column spaces of $\mathbf{A}$ *and* $\mathbf{B}$ *are orthogonal.*

Proof: See Appendix B. ■

It is easy to see that theorems 1 and 2 can be extended for the concatenation of multiple matrices. Thus, for (6), we have,

$$\sum_{c=1}^{C} ||\mathbf{T}\mathbf{Y}_c||_* - ||\mathbf{T}\mathbf{Y}||_* \geq 0. \tag{7}$$

Based on (7) and Theorem 2, the proposed objective function (6) reaches the minimum 0 if the column spaces of every pair of matrices are orthogonal after applying the learned transformation $\mathbf{T}$; or equivalently, (6) reaches the minimum 0 when the separation between every pair of subspaces is maximized after transformation, i.e., the smallest principal angle between subspaces equals $\frac{\pi}{2}$. Note that such improved separation is not obtained if the rank is used in the second term in (6), thereby further justifying the use of the nuclear norm instead.

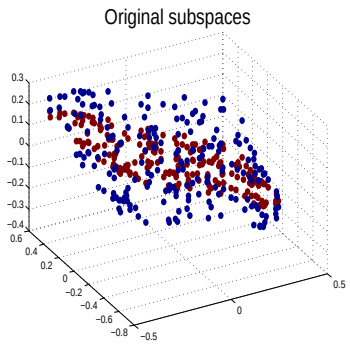
We have then, both intuitively and theoretically, justified the selection of the criteria (6) for learning the transform $\mathbf{T}$. We now illustrate the properties of the learned transformation $\mathbf{T}$ using synthetic examples in Figure 2 (real examples are presented in Section 5). Here we adopt a projected subgradient method described in Appendix C (though other modern nuclear norm optimization techniques could be considered, including recent real-time formulations Sprechmann et al. (2012)) to search for the transformation matrix T that minimizes (6). As shown in Figure 2, the learned transformation $\mathbf{T}$ via (6) maximizes the separation between every pair of subspaces towards $\frac{\pi}{2}$, and reduces the deviation of the data points to the true subspace when noise is present. Note that, comparing Figure 2c to Figure2d, the learned transformation using (6) maximizes the angle between subspaces, and the nuclear norm changes from $|[\mathbf{A},\mathbf{B}]|_* = 1.41$ to $|[\mathbf{A},\mathbf{B}]|_* = 1.95$ to make $|\mathbf{A}|_* + |\mathbf{B}|_* - |[\mathbf{A},\mathbf{B}]|_* \approx 0$; However, in both cases, where subspaces are independent, $rank([\mathbf{A},\mathbf{B}]) = 2$, and $rank(\mathbf{A}) + rank(\mathbf{B}) - rank([\mathbf{A},\mathbf{B}]) = 0$.

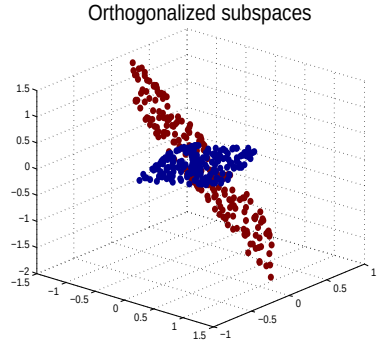
2.3 Comparisons with other Transformations

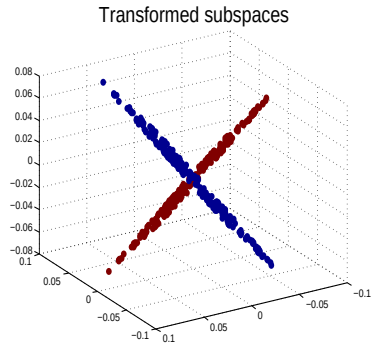
For independent subspaces, a transformation that renders them pairwise orthogonal can be obtained in a closed-form as follows: we take a basis $\mathbf{U}_c$ for the column space of $\mathbf{Y}_c$ for each subspace, form a matrix $\mathbf{U} = [\mathbf{U}_1, ..., \mathbf{U}_C]$, and then obtain the orthogonalizing transformation as $\mathbf{T} = (\mathbf{U}'\mathbf{U})^{-1}\mathbf{U}'$. To further elaborate the properties of our learned transformation, using synthetic examples, we compare with the closed-form orthogonalizing transformation in Figure 3 and with linear discriminant analysis (LDA) in Figure 4.

Two intersecting planes are shown in Figure 3a. Though subspaces here are neither independent nor disjoint, the closed-form orthogonalizing transformation still significantly increases the angle between the two planes towards $\frac{\pi}{2}$ in Figure 3b (note that the angle for the common line here is always 0). Note also that the closed-form orthogonalizing transformation is of size $r \times d$, where r is the sum of the dimension of each subspace, and we plot just the first 3 dimensions for visualization. Comparing to the orthogonalizing transformation, our leaned transformation in Figure 3c introduces similar subspace separation, but enables significantly reduced within subspace variations, indicated by the decreased nuclear norm values (close to 1). The same set of experiments with different samples per subspace are shown in the second row of Figure 3. Our formulation in (6) not only maximizes the separations between the different classes subspaces, but also simultaneously reduces the variations within the same class subspaces.

Our learned transformation shares a similar methodology with LDA, i.e., minimizing intra-class variation and maximizing inter-class separation. Two classes $\mathbf{Y}_+$ and $\mathbf{Y}_-$ are shown in Figure 4a, each class consisting of two lines. Our learned transformation in Figure 4c shows smaller intra-class variation than LDA in Figure 4b by merging two lines in each class, and simultaneously maximizes the angle between two classes towards $\frac{\pi}{2}$ (such two-class clustering and classification is critical for example for trees-based techniques Qiu and Sapiro (2014)). Note that we usually use LDA to reduce the data dimension to the number of classes minus 1; however, to better emphasize the distinction, we learn a $(d-1) \times d$ sized transformation matrix using both methods. The closed-form orthogonalizing transformation discussed above also gives higher intra-class variations as $|\mathbf{Y}_+|_* = 1.45$ and $|\mathbf{Y}_+|_* = 1.68$. Figure 4d shows an example of two non-linearly separable classes, i.e., two



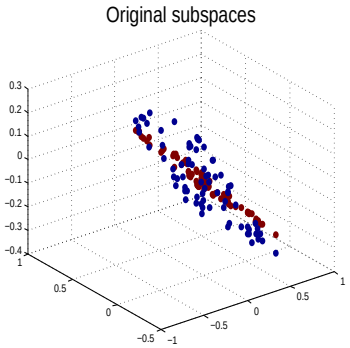


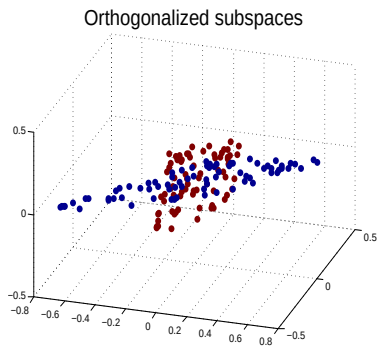


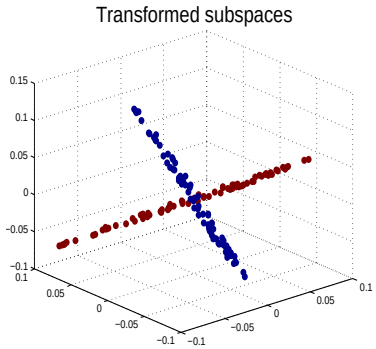
(a) 200 samples per plane, $\theta_{AB} = 0.31$, $|\mathbf{A}|_* = 1.91, |\mathbf{B}|_* = 1.88$.

(b) $\mathbf{T} = \begin{bmatrix} -0.36 & 0 & 0.10 \\ -1.4 & 0.09 & -2.98 \\ -0.92 & 0 & 1.54 \\ 1.06 & -1.12 & 2.63 \end{bmatrix}$; $\theta_{AB} = 1.41$, $|\mathbf{A}|_* = 1.61, \quad |\mathbf{B}|_* = 1.62$.

(c) $\mathbf{T} = \begin{bmatrix} 0.14 & 0.03 & 0.33 \\ -0.01 & 0.14 & -0.16 \\ 0.29 & -0.16 & 0.86 \end{bmatrix}$; $\theta_{AB} = 1.55$, $|\mathbf{A}|_* = 1.06, \quad |\mathbf{B}|_* = 1.06$.





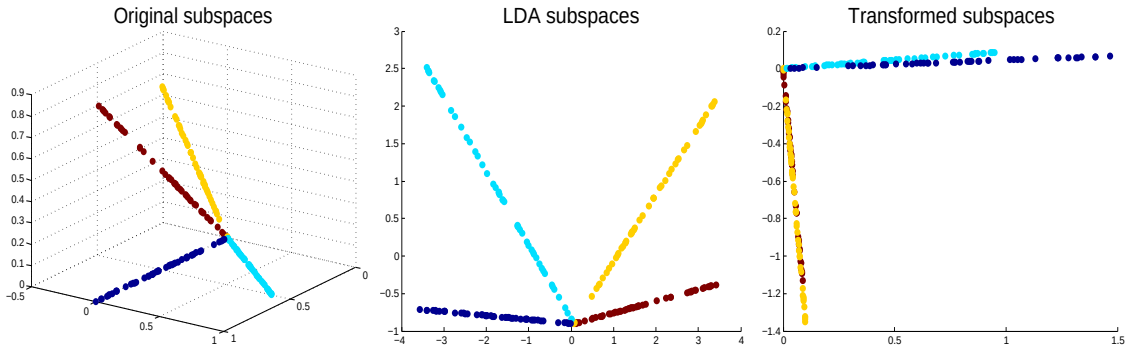


(d) 75 samples per plane, $\theta_{AB} = 0.31$, $|\mathbf{A}|_* = 1.92, |\mathbf{B}|_* = 1.81$.

(e) $\mathbf{T} = \begin{bmatrix} -1.62 & 0 & -1.94 \\ -0.50 & 0 & -2.38 \\ 0.25 & -0.50 & 1.81 \\ 1.12 & -1.50 & 2.37 \end{bmatrix}$; $\theta_{AB} = 1.04$, $|\mathbf{A}|_* = 1.75, \quad |\mathbf{B}|_* = 1.71$.

(f) $\mathbf{T} = \begin{bmatrix} 0.06 & -0.12 & 0.30 \\ -0.01 & 0.13 & -0.15 \\ 0.33 & -0.12 & 0.86 \end{bmatrix}$; $\theta_{AB} = 1.55$, $|\mathbf{A}|_* = 1.08, \quad |\mathbf{B}|_* = 1.07$.

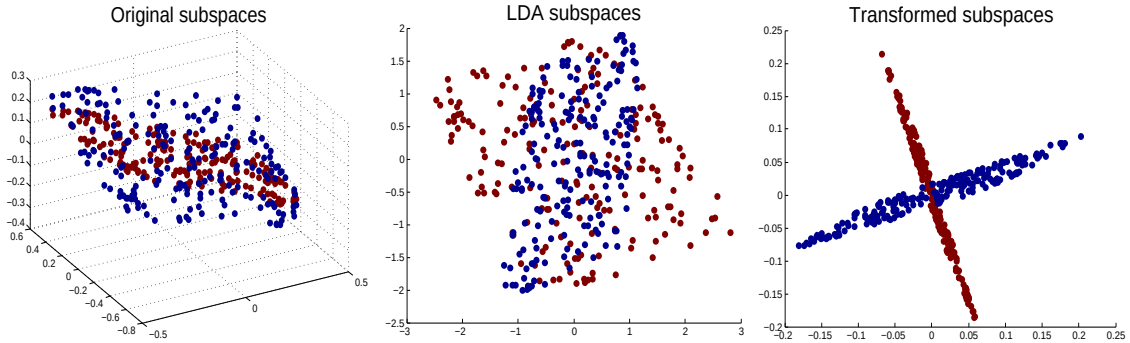
Figure 3: Comparisons with the closed-form orthogonalizing transformation. Two intersecting planes are shown in (a), and each plane contains 200 points. The closed-form orthogonalizing transformation significantly increase the angle between the two planes towards $\frac{\pi}{2}$ in (b). Our leaned transformation in (c) introduces similar subspace separation, but simultaneously enables significantly reduced within subspace variation, indicated by the smaller nuclear norm values (close to 1). The same set of experiments with 75 points per subspace are shown in the second row.



(a) Two classes $\{\mathbf{Y}_+, \mathbf{Y}_-\}$, $\mathbf{Y}_+ = \{\mathbf{A}(blue), \mathbf{B}(cyan)\}$, $\mathbf{Y}_- = \{\mathbf{C}(yellow), \mathbf{D}(red)\}$, $\begin{bmatrix}\theta_{AB} = 1.1, \theta_{AC} = 1.1, \theta_{AD} = 1.1,\\ \theta_{BC} = 1.3, \theta_{BD} = 1.4, \theta_{CD} = 0.5\end{bmatrix}$, $|\mathbf{Y}_+|_* = 1.58, \quad |\mathbf{Y}_-|_* = 1.27.$

(b) $\mathbf{T} = \begin{bmatrix}-3.64 & -1.95 & 5.98\\ 0.19 & 3.87 & 3.35\end{bmatrix}$; $\begin{bmatrix}\theta_{AB} = 0.7, \theta_{AC} = 0.78, \theta_{AD} = 0.2,\\ \theta_{BC} = 1.5, \theta_{BD} = 0.9, \theta_{CD} = 0.57\end{bmatrix}$, $|\mathbf{Y}_+|_* = 1.35, \quad |\mathbf{Y}_-|_* = 1.27.$

(c) $\mathbf{T} = \begin{bmatrix}1.47 & 0.26 & -0.73\\ 0.07 & 0.06 & -1.62\end{bmatrix}$; $\begin{bmatrix}\theta_{AB} = 0.04, \theta_{AC} = 1.54, \theta_{AD} = 1.54,\\ \theta_{BC} = 1.55, \theta_{BD} = 1.56, \theta_{CD} = 0.01\end{bmatrix}$, $|\mathbf{Y}_+|_* = 1.02, \quad |\mathbf{Y}_-|_* = 1.00.$



(d) Two classes $\{\mathbf{A}(\text{blue}), \mathbf{B}(\text{red})\}$, $\theta_{AB} = 0.31$, $|\mathbf{A}|_* = 1.91, |\mathbf{B}|_* = 1.88.$

(e) $\mathbf{T} = \begin{bmatrix}-0.54 & 2.60 & -9.51\\ 0.56 & -3.21 & -1.02\end{bmatrix}$; $\theta_{AB} = 0$, $|\mathbf{A}|_* = 1.52, \quad |\mathbf{B}|_* = 1.69.$

(f) $\mathbf{T} = \begin{bmatrix}0.49 & -0.11 & 1.27\\ -0.09 & 0.29 & -0.59\end{bmatrix}$; $\theta_{AB} = 1.57$, $|\mathbf{A}|_* = 1.08, \quad |\mathbf{B}|_* = 1.03.$

Figure 4: Comparisons with the linear discriminant analysis (LDA). Two classes $\mathbf{Y}_+$ and $\mathbf{Y}_-$ are shown in (a), each class consisting of two lines. We notice that our learned transformation (c) shows smaller intra-class variation than LDA in (b) by merging two lines in each class, and simultaneously maximizes the angle between two classes towards $\frac{\pi}{2}$ (such two-class clustering and classification is critical for example for trees-based techniques Qiu and Sapiro (2014)). (d) shows an example of two non-linearly separable classes, i.e., two intersecting planes, which cannot be improved by LDA in (e). However, our learned transformation in (f) prepares data to be separable using subspace clustering.

intersecting planes, which cannot be improved by LDA, as shown in Figure 4e. However, our learned transformation in Figure 4f prepares the data to be separable using subspace clustering. As shown in Qiu and Sapiro (2014), the property demonstrated above makes our learned transformation a better learner than LDA in a binary classification tree.

Lastly, we generated an interesting disjoint case: we consider three lines A, B and C on the same plane that intersect at the origin; the angles between them are $\theta_{AB} = 0.08$, $\theta_{BC} = 0.08$, and $\theta_{AC} = 0.17$. As the closed-form orthogonalizing approach is valid for independent subspaces, it fails by producing $\theta_{AB} = 0.005$, $\theta_{BC} = 0.005$, $\theta_{BC} = 0.01$. Our framework is not limited to that, even if additional theoretical foundations are yet to come. After our learned transformation, we have $\theta_{AB} = 1.20$, $\theta_{BC} = 1.20$, and $\theta_{AC} = 0.75$. We can make two immediate observations: First, all angles are significantly increased within the valid range of $[0, \frac{\pi}{2}]$. Second, $\theta_{AB}+\theta_{BC}+\theta_{AC} = \pi$ (we made the same two observations while repeating the experiments with different subspace angles). Though at this point we have no clean interpretation about how those angles are balanced when pair-wise orthogonality is not possible, we strongly believe that some theories are behind the above persistent observations and we are currently exploring this.

2.4 Discussions about Other Matrix Norms

We now discuss the advantages of replacing the rank function in (1) with the nuclear norm over other (popular) matrix norms, e.g., the induced 2-norm and the Frobenius norm.

Proposition 3 *Let* $\mathbf{A}$ *and* $\mathbf{B}$ *be matrices of the same row dimensions, and* $[\mathbf{A}, \mathbf{B}]$ *be the concatenation of* $\mathbf{A}$ *and* $\mathbf{B}$*, we have*

$$||[\mathbf{A}, \mathbf{B}]||_2 \leq ||\mathbf{A}||_2 + ||\mathbf{B}||_2,$$

with equality if at least one of the two matrices is zero.

Proposition 4 *Let* $\mathbf{A}$ *and* $\mathbf{B}$ *be matrices of the same row dimensions, and* $[\mathbf{A}, \mathbf{B}]$ *be the concatenation of* $\mathbf{A}$ *and* $\mathbf{B}$*, we have*

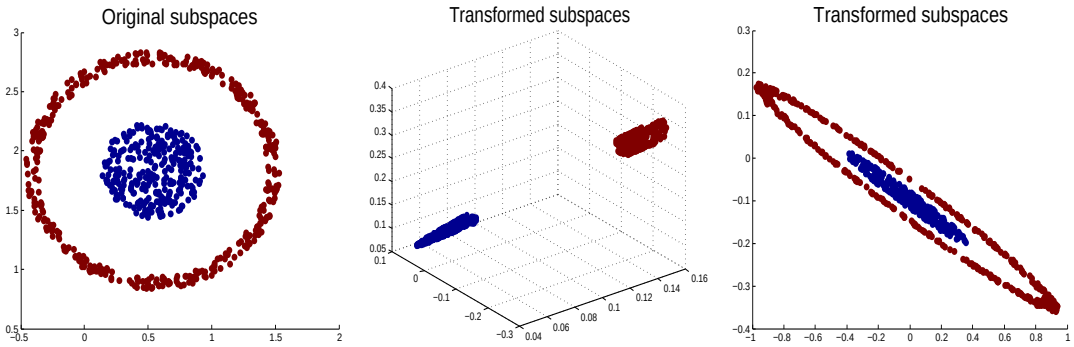
$$||[\mathbf{A}, \mathbf{B}]||_F \leq ||\mathbf{A}||_F + ||\mathbf{B}||_F,$$

with equality if and only if at least one of the two matrices is zero.

We choose the nuclear norm in (6) for two major advantages that are not so favorable in other (popular) matrix norms:

- The nuclear norm is the best convex approximation of the rank function Fazel (2002), which helps to reduce the variation within the subspaces (first term in (6));
- The objective function (6) is optimized when the distance between every pair of subspaces is maximized after transformation, which helps to introduce separations between the subspaces.

Note that (1), which is based on the rank, reaches the minimum when subspaces are independent but not necessarily maximally distant. Propositions 3 and 4 show that the property of the nuclear norm in Theorem 1 holds for the induced 2-norm and the Frobenius norm. However, if we replace the rank function in (1) with the induced 2-norm norm or the Frobenius norm, the objective function is minimized at the trivial solution $\mathbf{T} = 0$, which is prevented by the normalization condition $||\mathbf{T}||_2 = 1$.



(a) (b) (c)

Figure 5: A synthetic example illustrating the kernelized transformation learning. (a) is transformed to (b) with an RBF kernel applied, and to (c) without kernel.

2.5 Online Learning Low-rank Transformations

When data $\mathbf{Y}$ is big, we use an online algorithm to learn the low-rank transformation $\mathbf{T}$:

- We first randomly partition the data set $\mathbf{Y}$ into B mini-batches;
- Using mini-batch subgradient descent, a variant of stochastic subgradient descent, the subgradient in Appendix C is approximated by a sum of subgradients obtained from each mini-batch of samples,

$$\mathbf{T}^{(t+1)} = \mathbf{T}^{(t)} - \nu \sum_{b=1}^{B} \Delta \mathbf{T}_b, \tag{8}$$

  where $\Delta \mathbf{T}_b$ is obtained using only data points in the b-th mini-batch;
- Starting with the first mini-batch, we learn the subspace transformation $\mathbf{T}_b$ using data only in the b-th mini-batch, with $\mathbf{T}_{b-1}$ as warm restart.

2.6 Subspace Transformation with Compression

Given data $\mathbf{Y} \subseteq \mathbb{R}^d$, so far, we considered a square linear transformation $\mathbf{T}$ of size $d \times d$. If we devise a "fat" linear transformation $\mathbf{T}$ of size $r \times d$, where $(r < d)$, we enable dimension reduction along with transformation. This connects the proposed framework with the literature on compressed sensing, though the goal here is to learn a "sensing" matrix $\mathbf{T}$ for subspace classification and not for reconstruction Carson et al. (2012). The nuclear-norm minimization provides a new metric for such compressed sensing design (or compressed feature learning) paradigm. Results with this reduced dimensionality will be presented in Section 5.

2.7 Kernelized Transformation

The linear transformation suggested above shows effective when data approximately lie in linear subspaces. To improve the ability in handling more generic data, we can further map

data points into an inner product space prior to learning the transformation. Given a data point $\mathbf{y}$, we create a nonlinear map $\mathcal{K}(\mathbf{y}) = (\kappa(\mathbf{y}, \mathbf{y}_1); ...; \kappa(\mathbf{y}, \mathbf{y}_n))$ by computing the inner product between $\mathbf{y}$ and a fixed set of n points $\{\mathbf{y}_1, ..., \mathbf{y}_n\}$ randomly drawn from the training set. The inner products are computed via the kernel function, $\kappa(\mathbf{y}, \mathbf{y}_i) = \varphi(\mathbf{y})'\varphi(\mathbf{y}_i)$, which has to satisfy the Mercer conditions; note that no explicit representation for φ is required. Examples of kernel functions include polynomial kernels $\kappa(\mathbf{y}, \mathbf{y}_i) = (\mathbf{y}'\mathbf{y}_i + p)^q$ (with p and q being constants), and radial basis function (RBF) kernels $\kappa(\mathbf{y}, \mathbf{y}_i) = \exp(-\frac{||\mathbf{y}-\mathbf{y}_i||_2^2}{2\sigma^2})$ with variance σ^2. Given a set of data points $\mathbf{Y}$, the set of mapped data is denoted as $\mathcal{K}(\mathbf{Y}) \subseteq \mathbb{R}^n$. We now learn an $n \times n$ kernelized transformation $\mathbf{T}$ minimizing

$$\min_{\mathbf{T}} \sum_{c=1}^{C} ||\mathbf{T}\mathcal{K}(\mathbf{Y}_c)||_* - ||\mathbf{T}\mathcal{K}(\mathbf{Y})||_*, \quad \text{s.t. } ||\mathbf{T}||_2 = 1. \tag{9}$$

Figure 5 shows a synthetic example illustrating the kernelized transformation learning, where a 256-dimensional RBF kernel is applied.

3. Subspace Clustering using Low-rank Transformations

We now move from classification, where we learned the transform from training labeled data, to clustering, where no training data is available. In particular, we address the *subspace clustering* problem, meaning to partition the data set $\mathbf{Y}$ into C clusters corresponding to their underlying subspaces. We first present a general procedure to enhance the performance of existing subspace clustering methods in the literature. Then we further propose a specific fast subspace clustering technique to fully exploit the low-rank structure of (learned) transformed subspaces.

3.1 A Learned Robust Subspace Clustering (LRSC) Framework

In clustering tasks, the data labeling is of course not known beforehand in practice. The proposed algorithm, Algorithm 1, iterates between two stages: In the first assignment stage, we obtain clusters using any subspace clustering methods, e.g., SSC (Elhamifar and Vidal, 2013), LSA (Yan and Pollefeys, 2006), LBF (Zhang et al., 2012). In particular, in this paper we often use the new improved technique introduced in Section 3.2. In the second update stage, based on the current clustering result, we compute the optimal subspace transformation that minimizes (6). The algorithm is repeated until the clustering assignments stop changing.

The LRSC algorithm is a general procedure to enhance the performance of any subspace clustering methods, and part of the beauty of the proposed model is that it can be applied to any such algorithm, and even beyond (Qiu and Sapiro, 2014). We don't enforce an overall objective function at the present form for such versatility purpose.

To study convergence, one way is to adopt the subspace clustering method for the LRSC assignment step by optimizing the same LRSC update criterion (6): given the cluster assignment and the transformation $\mathbf{T}$ at the current LRSC iteration, we take a point $\mathbf{y}_i$ out of its current cluster (keep the rest assignments no change) and place it into a cluster $\mathbf{Y}_c$ that minimize $\sum_{c=1}^{C} ||\mathbf{T}\mathbf{Y}_c||_*$. We iteratively perform this for all points, and then update

$\mathbf{T}$ using current $\mathbf{T}$ as warm restart. In this way, we decrease (or keep) the overall objective function (6) after each LRSC iteration.

However, the above approach is computational expensive and only allow one specific subspace clustering method. Thus, in the present implementation, an overall objective function of the type that the LRSC algorithm optimizes can take a form such as

$$\arg\min_{\mathbf{T},\{\mathcal{S}_c\}_{c=1}^{C}} \sum_{c=1}^{C} \sum_{\mathbf{y}_i \in \mathcal{S}_c} ||\mathbf{T}\mathbf{y}_i - P_{\mathbf{T}\mathbf{Y}_c}\mathbf{T}\mathbf{y}_i||_2^2 + \lambda[\sum_{c=1}^{C} ||\mathbf{T}\mathbf{Y}_c||_* - ||\mathbf{T}\mathbf{Y}||_*], \quad \text{s.t.} ||\mathbf{T}||_2 = 1, \tag{10}$$

where $\mathbf{Y}_c$ denotes the set of points $\mathbf{y}_i$ in the c-th subspace $\mathcal{S}_c$, and $P_{\mathbf{T}\mathbf{Y}_c}$ denotes the projection onto $\mathbf{T}\mathbf{Y}_c$. The LRSC iterative algorithm optimize (10) through alternative minimization (with a similar form as the popular k-means, but with a different data model and with the learned transform). While formally studying its convergence is the subject of future research, the experimental validation presented already demonstrates excellent performance, with LRSC just one of the possible applications of the proposed learned transform.

In all our experiments, we observe significant clustering error reduction in the first few LRSC iterations, and the proposed LRSC iterations enable significantly cleaner subspaces for all subspace clustering benchmark data in the literature. The intuition behinds the observed empirical convergence is that the update step in each LRSC iteration decreases the second term in (10) to a small value close to 0 as discussed in Section 2; at the same time, the updated transformation tends to reduce the intra-subspace variation, which further reduces the first cluster deviation term in (10) even with assignments derived from various subspace clustering methods.

Input: A set of data points $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^{N} \subseteq \mathbb{R}^d$ in a union of C subspaces.
Output: A partition of $\mathbf{Y}$ into C disjoint clusters $\{\mathbf{Y}_c\}_{c=1}^{C}$ based on underlying subspaces.
begin
1. Initial a transformation matrix $\mathbf{T}$ as the identity matrix ;
repeat
Assignment stage:
2. Assign points in $\mathbf{TY}$ to clusters with any subspace clustering methods, e.g., the proposed R-SSC;
Update stage:
3. Obtain transformation $\mathbf{T}$ by minimizing (6) based on the current clustering result ;
until *assignment convergence*;
4. Return the current clustering result $\{\mathbf{Y}_c\}_{c=1}^{C}$;
end

Algorithm 1: Learning a robust subspace clustering (LRSC) framework.

3.2 Robust Sparse Subspace Clustering (R-SSC)

Though Algorithm 1 can adopt any subspace clustering methods, to fully exploit the low-rank structure of the learned transformed subspaces, we further propose the following specific technique for the clustering step in the LRSC framework, called Robust Sparse Subspace Clustering (R-SSC):

1. For the transformed subspaces, we first recover their low-rank representation $\mathbf{L}$ by performing a low-rank decomposition (11), e.g., using RPCA (Candès et al., 2011),[1]

$$\arg\min_{\mathbf{L},\mathbf{S}} ||\mathbf{L}||_* + \beta||\mathbf{S}||_1 \quad \text{s.t. } \mathbf{TY} = \mathbf{L} + \mathbf{S}. \tag{11}$$

2. Each transformed point $\mathbf{Ty}_i$ is then sparsely decomposed over $\mathbf{L}$,

$$\arg\min_{\mathbf{x}_i} \|\mathbf{Ty}_i - \mathbf{L}\mathbf{x}_i\|_2^2 \quad \text{s.t. } \|\mathbf{x}_i\|_0 \leq K, \tag{12}$$

   where K is a predefined sparsity value ($K > d$). As explained in Elhamifar and Vidal (2013), a data point in a linear or affine subspace of dimension d can be written as a linear or affine combination of d or $d+1$ points in the same subspace. Thus, if we represent a point as a linear or affine combination of all other points, a sparse linear or affine combination can be obtained by choosing d or $d+1$ nonzero coefficients.
3. As the optimization process for (12) is computationally demanding, we further simplify (12) using Local Linear Embedding (Roweis and Saul, 2000; Wang et al., 2010). Each transformed point $\mathbf{Ty}_i$ is represented using its K Nearest Neighbors (NN) in $\mathbf{L}$, which are denoted as $\mathbf{L}_i$,

$$\arg\min_{\mathbf{x}_i} \|\mathbf{Ty}_i - \mathbf{L}_i\mathbf{x}_i\|_2^2 \quad \text{s.t. } \mathbf{1}'\mathbf{x}_i = 1. \tag{13}$$

   Let $\bar{\mathbf{L}}_i = \mathbf{L}_i - \mathbf{1}\mathbf{Ty}_i^T$. $\mathbf{x}_i$ can then be efficiently obtained in closed form (Saul and Roweis, 2000),

$$\mathbf{x}_i = \bar{\mathbf{L}}_i\bar{\mathbf{L}}_i^T \setminus \mathbf{1},$$

   where $\mathbf{x} = \mathbf{A} \setminus \mathbf{B}$ solves the system of linear equations $\mathbf{Ax} = \mathbf{B}$, and then we rescale $\mathbf{x}_i$ so that $\mathbf{1}'\mathbf{x}_i = 1$. As suggested in Roweis and Saul (2000), if the correlation matrix $\bar{\mathbf{L}}_i\bar{\mathbf{L}}_i^T$ is nearly singular, it can be conditioned by adding a small multiple of the identity matrix. From experiments, we observe this simplification step dramatically reduces the running time, without sacrificing the accuracy.
4. Given the sparse representation $\mathbf{x}_i$ of each transformed data point $\mathbf{Ty}_i$, we denote the sparse representation matrix as $\mathbf{X} = [\mathbf{x}_1 \dots \mathbf{x}_N]$. It is noted that $\mathbf{x}_i$ is written as an N-sized vector with no more than $K << N$ non-zero values (N being the total number of data points). The pairwise affinity matrix is now defined as $\mathbf{W} = |\mathbf{X}| + |\mathbf{X}^T|$, and the subspace clustering is obtained using spectral clustering (Luxburg, 2007).

Based on experimental results presented in Section 5, the proposed R-SSC outperforms state-of-the-art subspace clustering techniques, in both accuracy and running time, e.g., about 500 times faster than the original SSC using the implementation provided in Elhamifar and Vidal (2013). Performance is further enhanced when R-SCC is used as an internal step of LRSC in Algorithm 1.

1. Note that while the learned transform $\mathbf{T}$ encourages low-rank in each sub-space, outliers might still exist. Moreover, during the iterations in Algorithm 1, the intermediate learned $\mathbf{T}$ is not yet the desired one. This justifies the incorporation of this further low-rank decomposition.

4. Classification using Single or Multiple Low-rank Transformations

In Section 2, learning one global transformation over all classes has been discussed, and then incorporated into a clustering framework in Section 3. The availability of data labels for training enables us to consider instead learning individual class-based linear transformation. The problem of class-based linear transformation learning can be formulated as

$$\arg\min_{\{\mathbf{T}_c\}_{c=1}^{C}} \sum_{c=1}^{C} [||\mathbf{T}_c\mathbf{Y}_c||_* - \lambda||\mathbf{T}_c\mathbf{Y}_{\neg c}||_*], \tag{14}$$

where $\mathbf{T}_c \in \mathbb{R}^{d\times d}$ denotes the transformation for the c-th class, $\mathbf{Y}_{\neg c} = \mathbf{Y} \setminus \mathbf{Y}_c$ denotes all data except the c-th class, and λ is a positive balance parameter.

When a global transformation matrix $\mathbf{T}$ is learned, we can perform classification in the transformed space by simply considering the transformed data $\mathbf{TY}$ as the new features. For example, when a Nearest Neighbor (NN) classifier is used, a testing sample $\mathbf{y}$ uses $\mathbf{Ty}$ as the feature and searches for nearest neighbors among $\mathbf{TY}$.

To fully exploit the low-rank structure of the transformed data, we propose to perform classification through the following procedure:

- For the c-th class, we first recover its low-rank representation $\mathbf{L}_c$ by performing low-rank decomposition (15), e.g., using RPCA (Candès et al., 2011):[2]

$$\arg\min_{\mathbf{L}_c,\mathbf{S}_c} ||\mathbf{L}_c||_* + \beta||\mathbf{S}_c||_1 \quad \text{s.t. } \mathbf{TY}_c = \mathbf{L}_c + \mathbf{S}_c. \tag{15}$$

- Each testing image $\mathbf{y}$ will then be assigned to the low-rank subspace $\mathbf{L}_c$ that gives the minimal reconstruction error through sparse decomposition, e.g., using OMP (Pati et al., Nov. 1993):

$$\arg\min_{\mathbf{x}} \|\mathbf{Ty} - \mathbf{L}_i\mathbf{x}\|_2^2 \quad \text{s.t. } \|\mathbf{x}\|_0 \leq T, \tag{16}$$

  where T is a predefined sparsity value.

When class-based transformations $\{\mathbf{T}_c\}_{c=1}^{C}$ are learned, we perform recognition in a similar way. However, now we apply all the learned transforms $\mathbf{T}_c$ to each testing data point and then pick the best one using the same criterion of minimal reconstruction error through sparse decomposition (16).

5. Experimental Evaluation

This section first presents experimental evaluations on subspace clustering using three public data sets (standard benchmarks): the MNIST handwritten digit data set, the Extended YaleB face data set (Georghiades et al., 2001) and the Hopkins 155 database of motion segmentation. The MNIST data set consists of 8-bit gray scale handwritten digit images of "0" through "9" and 7000 examples for each class. The Extended YaleB face data set

2. Note that this is done only once and can be considered part of the training stage. As before, this further low-rank decomposition helps to handle outliers not addressed by the learned transform.

contains 38 subjects with near frontal pose under 64 lighting conditions. All the images are resized to 16×16. The classical Hopkins 155 database of motion segmentation, which is available at `http://www.vision.jhu.edu/data/hopkins155`, contains 155 video sequences along with extracted feature trajectories, where 120 of the videos have two motions and 35 of the videos have three motions.

Subspace clustering methods compared are SSC (Elhamifar and Vidal, 2013), LSA (Yan and Pollefeys, 2006), and LBF (Zhang et al., 2012). Based on the studies in Elhamifar and Vidal (2013), Vidal (2011) and Zhang et al. (2012), these three methods exhibit state-of-the-art subspace clustering performance. We adopt the LSA and SSC implementations provided in Elhamifar and Vidal (2013) from `http://www.vision.jhu.edu/code/`, and the LBF implementation provided in Zhang et al. (2012) from `http://www.ima.umn.edu/~zhang620/lbf/`. We adopt similar setups as described in Zhang et al. (2012) for experiments on subspace clustering.

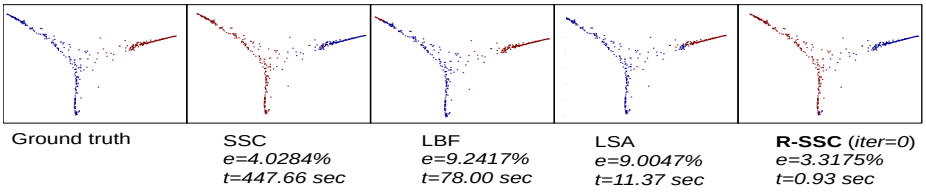
This section then presents experimental evaluations on classification using two public face data sets: the CMU PIE data set (Sim et al., 2003) and the Extended YaleB data set. The PIE data set consists of 68 subjects imaged simultaneously under 13 different poses and 21 lighting conditions. All the face images are resized to 20×20. We adopt a NN classifier unless otherwise specified.

5.1 Subspace Clustering with Illustrative Examples

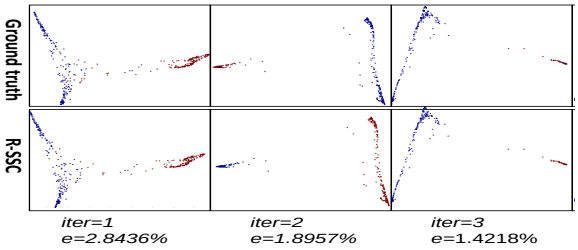
For illustration purposes, we conduct the first set of experiments on a subset of the MNIST data set. We adopt a similar setup as described in Zhang et al. (2012), using the same sets of 2 or 3 digits, and randomly choose 200 images for each digit. We set the sparsity value $K = 6$ for R-SSC, and perform 100 iterations for the subgradient updates while learning the transformation on subspaces. The subgradient update step was $\nu = 0.02$ (see Appendix C for details on the projected subgradient optimization algorithm).

Unless otherwise stated, we do not perform dimension reduction, such as PCA or random projections, to preprocess the data, thereby further saving computations (please note that the learned transform can itself reduce dimensions if so desired, see Section 5.8). In the literature, e.g., Elhamifar and Vidal (2013), Vidal (2011) and Zhang et al. (2012), projection to a very low dimension is usually performed to enhance the clustering performance. However, it is often not obvious how to determine the correct projection dimension for real data, and many subspace clustering methods show sensitive to the choice of the projection dimension. This dimension reduction step is not needed in the framework proposed here.

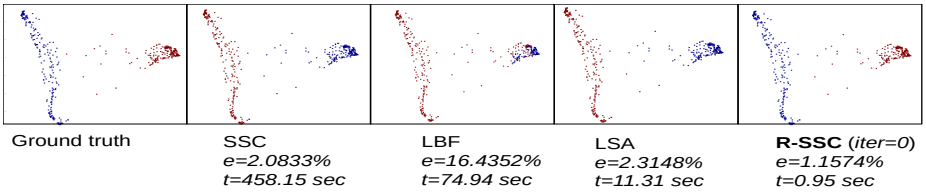
Figure 6 shows the misclassification rate (e) and running time (t) on clustering subspaces of two digits. The misclassification rate is the ratio of misclassified points to the total number of points, i.e., the ratio of points that were assigned to the wrong cluster. For visualization purposes, the data are plotted with the dimension reduced to 2 using Laplacian Eigenmaps Belkin and Niyogi (2003). Different clusters are represented by different colors and the ground truth is plotted using the true cluster labels. The proposed R-SSC outperforms state-of-the-art methods, both in terms of clustering accuracy and running time. The clustering error of R-SSC is further reduced using the proposed LRSC framework in Algorithm 1 through the learned low-rank subspace transformation. The clustering converges after about 3 LRSC iterations. The learned transformation not only recovers a low-rank



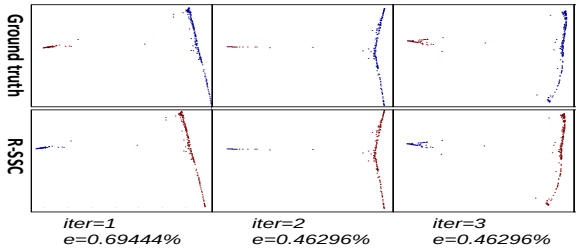
(a) Original subspaces for digits {1, 2}.



(b) Transformed subspaces for digits {1, 2}.

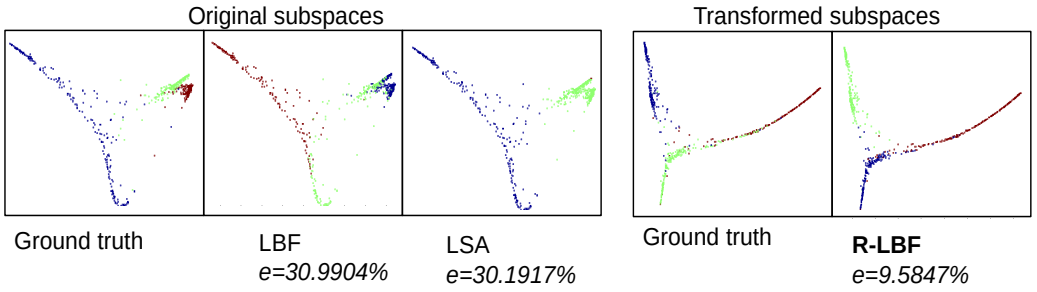


(c) Original subspaces for digits {1, 7}.

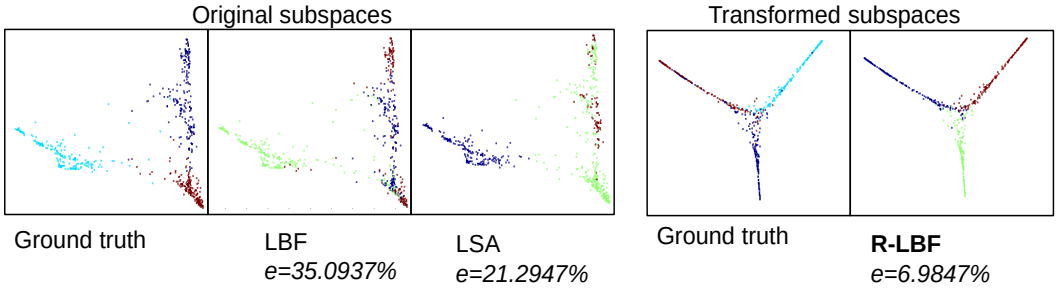


(d) Transformed subspaces for digits {1, 7}.

Figure 6: Misclassification rate (e) and running time (t) on clustering 2 digits. Methods compared are SSC Elhamifar and Vidal (2013), LSA Yan and Pollefeys (2006), and LBF Zhang et al. (2012). For visualization, the data are plotted with the dimension reduced to 2 using Laplacian Eigenmaps Belkin and Niyogi (2003). Different clusters are represented by different colors and the *ground truth* is plotted with the true cluster labels. *iter* indicates the number of LRSC iterations in Algorithm 1. The proposed R-SSC outperforms state-of-the-art methods in terms of both clustering accuracy and running time, e.g., about 500 times faster than SSC. The clustering performance of R-SSC is further improved using the proposed LRSC framework. Note how the data is clearly clustered in clean subspaces in the transformed domain (best viewed zooming on screen).



(a) Digits {1, 2, 3}.



(b) Digits {2, 4, 8}.

Figure 7: Misclassification rate (e) on clustering 3 digits. Methods compared are LSA Yan and Pollefeys (2006) and LBF Zhang et al. (2012). LBF is adopted in the proposed LRSC framework and denoted as R-LBF. After convergence, R-LBF significantly outperforms state-of-the-art methods.

Subsets	[0:1]	[0:2]	[0:3]	[0:4]	[0:5]	[0:6]	[0:7]	[0:8]
C	2	3	4	5	6	7	8	9
LSA	0.47	47.57	36.73	30.90	40.46	48.13	39.87	44.03
LBF	0.47	23.62	29.19	51.37	48.99	53.01	39.87	38.79
LRSC	**0**	**3.88**	**3.89**	**5.31**	**14.04**	**13.79**	**14.50**	**16.05**

Table 1: Misclassification rate ($e\%$) on clustering different numbers of digits in the MNIST data set, $[0 : c]$ denotes the subset of $c + 1$ digits from digit 0 to c. We randomly pick 100 samples per digit. For all cases, the proposed LRSC method significantly outperforms state-of-the-art methods.

structure for data from the same subspace, but also increases the separations between the subspaces for more accurate clustering.

Figure 7 shows misclassification rate (e) on clustering subspaces of three digits. Here we adopt LBF in our LRSC framework, denoted as Robust LBF (R-LBF), to illustrate that the performance of existing subspace clustering methods can be enhanced using the proposed LRSC algorithm. After convergence, R-LBF, which uses the proposed learned subspace transformation, significantly outperforms state-of-the-art methods.

Table 1 shows the misclassification rate on clustering different number of digits, $[0 : c]$ denotes the subset of $c+1$ digits from digit 0 to c. We randomly pick 100 samples per digit to compare the performance when a fewer number of data points per class are present. For all cases, the proposed LRSC method significantly outperforms state-of-the-art methods.

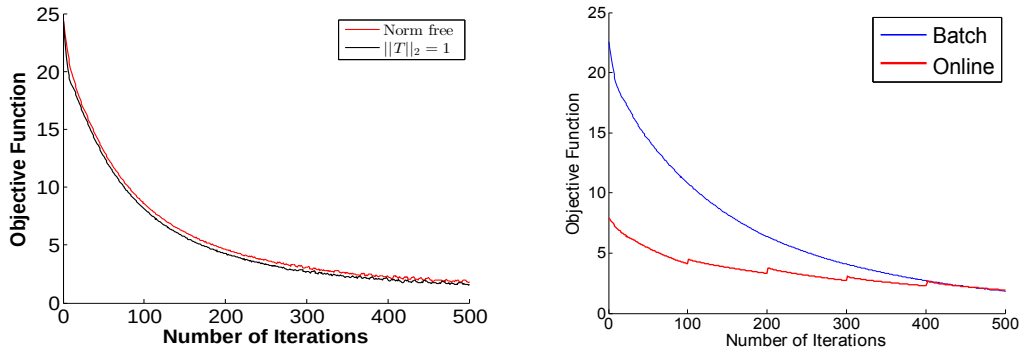
5.1.1 Online vs. Batch Learning

In this set of experiments, we use digits $\{1, 2\}$ from the MNIST data set. We select 1000 images for each digit, and randomly partition them into 5 mini-batches. We first perform one iteration of LRSC in Algorithm 1 over all selected data with and without the norm constraint. As shown in Figure 8a, we both observe empirical convergence for subspace transformation learning via (6) using the projected subgradient method presented in Appendix C.

Starting with the first mini-batch, we then perform one iteration of LRSC over one mini-batch a time, with the subspace transformation learned from the previous mini-batch as warm restart. We adopt here 100 iterations for the subgradient descent updates. As shown in Figure 8b, we observe similar empirical convergence for online transformation learning. To converge to the same objective function value, it takes 131.76 sec. for online learning and 700.27 sec. for batch learning.

5.2 Application to Face Clustering

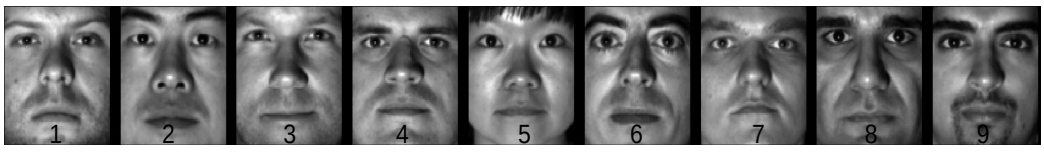
In the Extended YaleB data set, each of the 38 subjects is imaged under 64 lighting conditions, shown in Figure 9a. Under the assumption of Lambertian reflectance, face images of each subject under different lighting conditions can be accurately approximated with a 9-dimensional linear subspace (Basri and Jacobs, 2003). We conduct the face clustering experiments on the first 9 subjects shown in Figure 9b. We set the sparsity value $K = 10$



(a) Batch learning. (b) Online vs. batch learning ($\gamma = 1$).

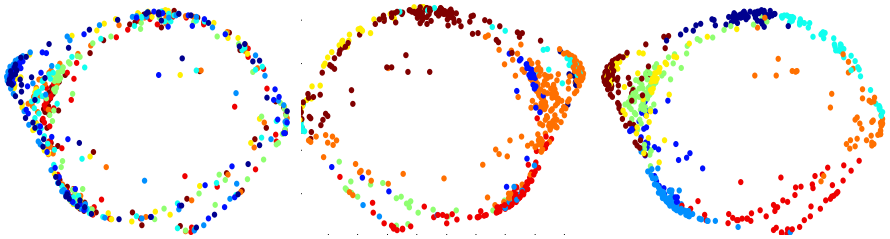
Figure 8: Convergence of the objective function (6) using online and batch learning for subspace transformation. We always observe empirical convergence for both online and batch learning. In (a), we learn with and without the norm constraint respectively. More discussions on convergence can be found in Appendix C. In (b), to converge to the same objective function value, it takes 131.76 sec. for online learning and 700.27 sec. for batch learning.

(a) Example illumination conditions.

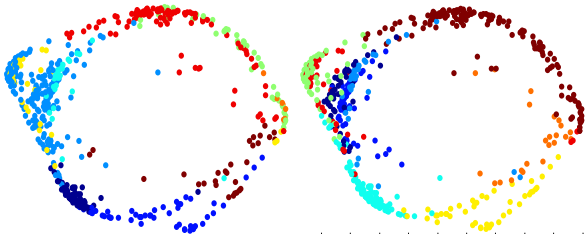


(b) Example subjects.

Figure 9: The extended YaleB face data set.

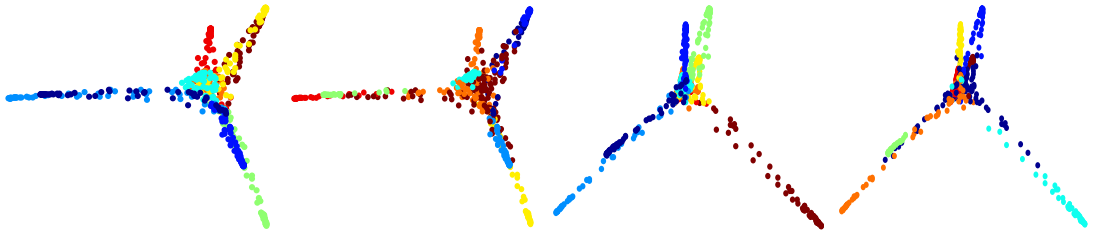


(a) Ground truth. (b) SSC, $e = 71.25\%$, $t = 714.99$ *sec.* (c) LBF, $e = 76.37\%$, $t = 460.76$ *sec.*

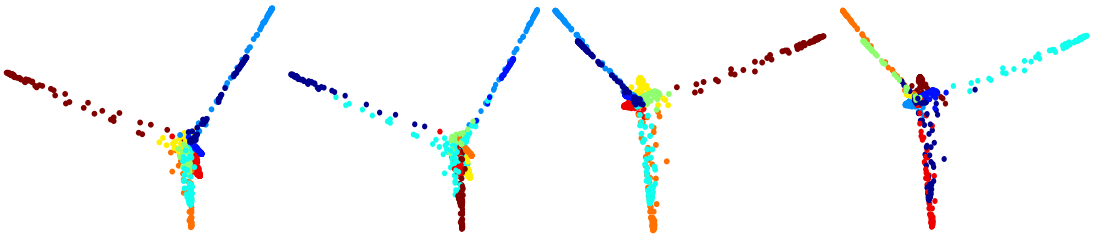


(d) LSA, $e = 71.96\%$, $t = 22.57$ *sec.* (e) R-SSC, $e = \mathbf{67.37}\%$, $t = \mathbf{1.83}$ *sec.*

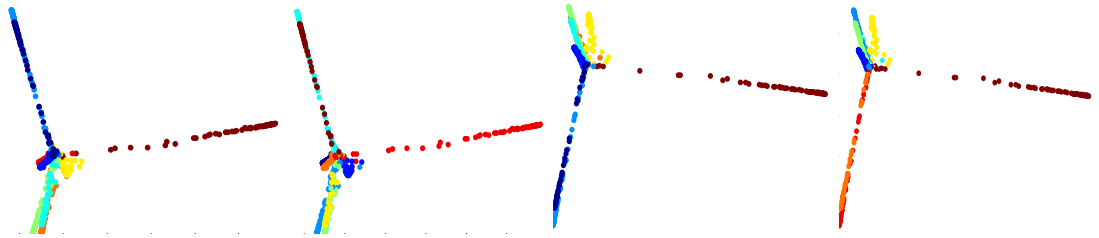
Figure 10: Misclassification rate (e) and running time (t) on clustering 9 subjects using different subspace clustering methods. The proposed R-SSC outperforms state-of-the-art methods both in accuracy and running time. This is further improved using the learned transform, LRSC reduces the error to 4.94%, see Figure 11.



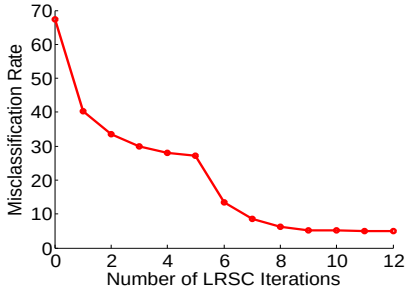
(a) Ground truth (iter=1).(b) $e = 40.39\%$ (iter=1).(c) Ground truth (iter=2).(d) $e = 33.51\%$ (iter=2).



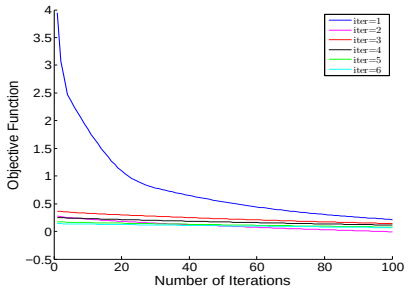
(e) Ground truth (iter=3).(f) $e = 29.98\%$ (iter=3).(g) Ground truth (iter=6).(h) $e = 13.40\%$ (iter=6).



(i) Ground truth (iter=8). (j) $e = 6.17\%$ (iter=8). (k) Ground truth (iter=12). (l) $e = \mathbf{4.94}\%$ (iter=12).

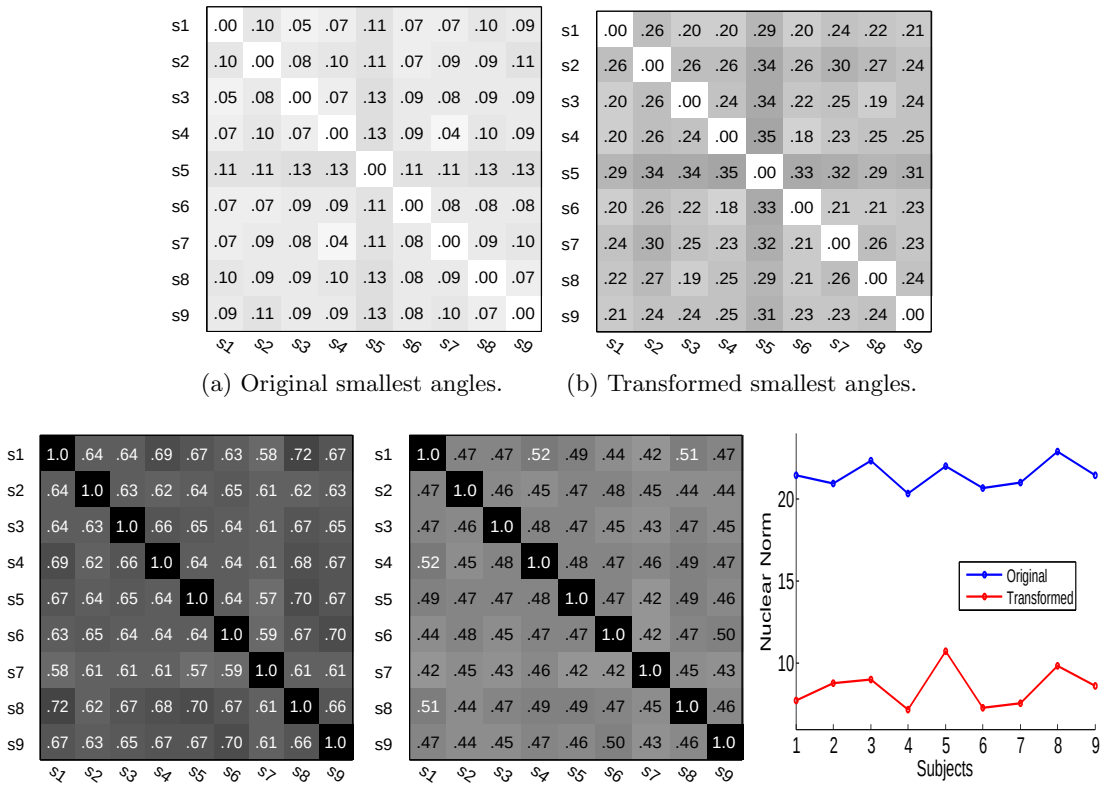


(m) Misclassification rate.



(n) Convergence of $\mathbf{T}$ updating

Figure 11: Misclassification rate (e) on clustering 9 subjects using the proposed LRSC framework. We adopt the proposed R-SSC technique for the clustering step. With the proposed LRSC framework, the clustering error of R-SSC is further reduced significantly, e.g., from 67.37% to 4.94% for the 9-subject case. Note how the classes are clustered in clean subspaces in the transformed domain.



(a) Original smallest angles. (b) Transformed smallest angles.

(c) Original mean cosine angles. (d) Transformed mean cosine angles. (e) Subspace nuclear norm.

Figure 12: The smallest and mean principal angles between pairs of 9 subject subspaces and the nuclear norms of 9 subject subspaces before and after transformation. Note that each entry in (a) and (b) denotes the smallest principal angle, and each entry in (c) and (d) denotes the average cosine over all principal angles. We observe that the learned subspace transformation increases the angles between subspaces and also reduces the nuclear norms of subspaces. Overall, the average smallest principal angles between subspaces increased from 0.09 to 0.26, and the average subspace nuclear norm decreased from 21.43 to 8.53.

Subsets	[1:10]	[1:15]	[1:20]	[1:25]	[1:30]	[1:38]
C	10	15	20	25	30	38
LSA	78.25	82.11	84.92	82.98	82.32	84.79
LBF	78.88	74.92	77.14	78.09	78.73	79.53
LRSC	**5.39**	**4.76**	**9.36**	**8.44**	**8.14**	**11.02**

Table 2: Misclassification rate ($e\%$) on clustering different number of subjects in the Extended YaleB face data set, $[1:c]$ denotes the first c subjects in the data set. For all cases, the proposed LRSC method significantly outperforms state-of-the-art methods.

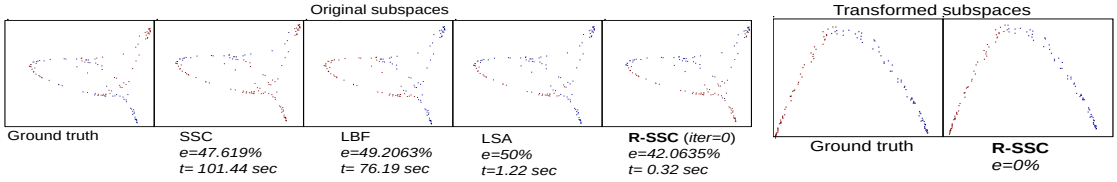
Methods	Misclassification (%)
orthogonalizing	61.36
LDA	9.77
Proposed	**5.47**

Table 3: Misclassification rate ($e\%$) on clustering 38 subjects in the Extended YaleB data set using supervised transformation learning. The proposed transformation learning outperforms both the closed-form orthogonalizing transformation and LDA on clustering the transformed data.

for R-SSC, and perform 100 iterations for the subgradient descent updates while learning the transformation.

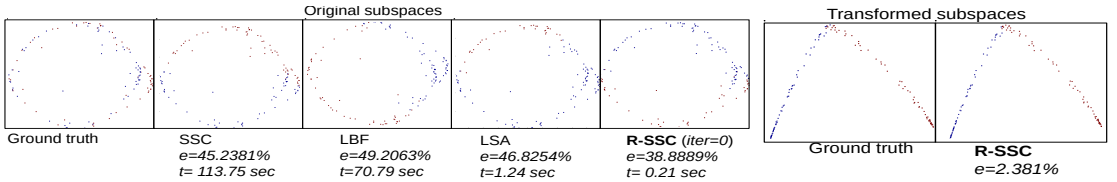
Figure 10 shows error rate (e) and running time (t) on clustering subspaces of 9 subjects using different subspace clustering methods. The proposed R-SSC techniques outperforms state-of-the-art methods both in accuracy and running time. As shown in Figure 11, using the proposed LRSC algorithm (that is, learning the transform), the misclassification errors of R-SSC are further reduced significantly, for example, from 67.37% to 4.94% for the 9 subjects. Figure 11n shows the convergence of the $\mathbf{T}$ updating step in the first few LRSC iterations. The dramatic performance improvement can be explained in Figure 12. We observe, as expected from the theory presented before, that the learned subspace transformation increases the distance (the smallest principal angle) between subspaces and, at the same time, reduces the nuclear norms of subspaces. More results on clustering subspaces of 2 and 3 subjects are shown in Figure 13.

Table 2 shows misclassification rate (e) on clustering subspaces of different number of subjects, $[1:c]$ denotes the first c subjects in the extended YaleB data set. For all cases, the proposed LRSC method significantly outperforms state-of-the-art methods. Note that without the low-rank decomposition step in (11), we obtain a misclassification rate 18.38% for clustering all 38 subjects in the Extended YaleB data set, which is slightly lower than the 11.02% reported in Table 2. Thus, pushing the subspaces apart through our learned transformation plays a major role here; and the robustness in the low-rank decomposition enhances the performance even further.



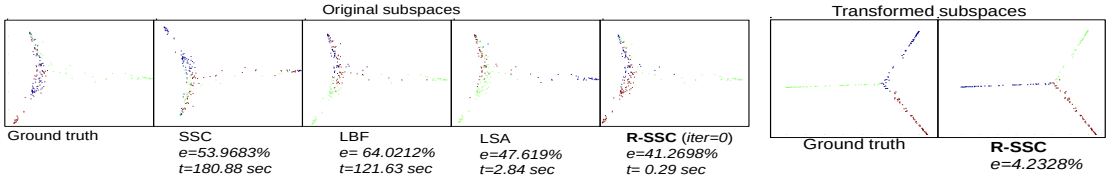
(a) Subjects {1, 2}.

(b) Subjects {1, 2}.



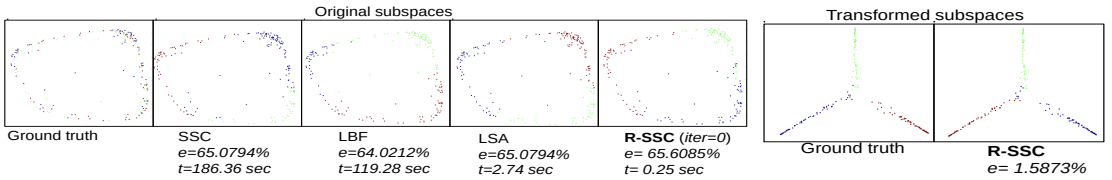
(c) Subjects {2, 3}.

(d) Subjects {2, 3}.



(e) Subjects {4, 5, 6}.

(f) Subjects {4, 5, 6}.



(g) Subjects {7, 8, 9}.

(h) Subjects {7, 8, 9}.

Figure 13: Misclassification rate (e) and running time (t) on clustering 2 and 3 subjects. The proposed R-SSC outperforms state-of-the-art methods both in accuracy and running time. With the proposed LRSC framework, the clustering error of R-SSC is further reduced significantly. Note how the classes are clustered in clean subspaces in the transformed domain (best viewed zooming on screen).

	Check		Traffic		Articulated		All	
	Mean	Median	Mean	Median	Mean	Median	Mean	Median
2-motion								
LSA	2.57	0.27	5.43	1.48	4.10	1.22	3.45	0.59
LBF	1.59	0	0.20	0	0.80	0	1.16	0
SSC	1.12	0	0.02	0	0.62	0	**0.82**	0
LRSC	1.19	0	0.23	0	0.88	0	0.92	0
3-motion								
LSA	5.80	1.77	25.07	23.79	7.25	7.25	9.73	2.33
LBF	4.57	0.94	0.38	0	2.66	2.66	3.63	0.64
SSC	2.97	0.27	0.58	0	1.42	0	2.45	0.2
LRSC	1.59	0	0.32	0	1.60	1.60	**1.34**	0

Table 4: Misclassification rate ($e\%$) on two motions and three motions segmentation in the Hopkins 155 data set. As shown in Vidal (2011); Zhang et al. (2012), the SSC method significantly outperforms all previous state-of-the-art methods on this data set. The proposed LRSC shows comparable results to SSC for two motions and outperforms SSC for three motions. Note that our method is orders of magnitude faster than SSC.

In Figure 3 and Figure 4, using synthetic examples, we previously compared our learned transformation with the closed-form orthogonalizing transformation and LDA. In Table 3, we further compare three transformations using real data. We perform supervised transformation learning on all 38 subjects in the Extended YaleB data set using three different transformation learning algorithms, and then perform subspace clustering on the transformed data. The proposed transformation learning significantly outperforms the other two methods.

5.3 Application to Motion Segmentation

The Hopkins 155 data set consists of three types of videos: checker, traffic and articulated, and 120 of the videos have two motions and 35 of the videos have three motions. The main task is to segment a video sequence of multiple rigidly moving objects into multiple spatiotemporal regions that correspond to different motions in the scene. This motion data set contains much cleaner subspace data than the digits and faces data evaluated above. To enable a fair comparison, we project the data into a lower dimensional subspace using PCA as explained in Vidal (2011); Zhang et al. (2012). Results on other comparing methods are taken from Vidal (2011). As shown in Vidal (2011); Zhang et al. (2012), the SSC method significantly outperforms all previous state-of-the-art methods on this data set. From Table 4, we can see that our method shows comparable results to SSC for two motions and outperforms SSC for three motions. Note that our method is orders of magnitude faster than SSC as discussed earlier.

Method	Accuracy (%)
D-KSVD Zhang and Li (2010)	94.10
LC-KSVD Jiang et al. (2011)	96.70
SRC Wright et al. (2009)	97.20
Original+NN	91.77
Class LRT+NN	97.86
Class LRT+OMP	92.43
Global LRT+NN	99.10
Global LRT+OMP	**99.51**

Table 5: Recognition accuracies (%) under illumination variations for the Extended YaleB data set. The recognition accuracy is increased from 91.77% to 99.10% by simply applying the learned low-rank transformation (LRT) matrix to the original face images.

5.4 Application to Face Recognition across Illumination

For the Extended YaleB data set, we adopt a similar setup as described in Jiang et al. (2011); Zhang and Li (2010). We split the data set into two halves by randomly selecting 32 lighting conditions for training, and the other half for testing. We learn a global low-rank transformation matrix from the training data.

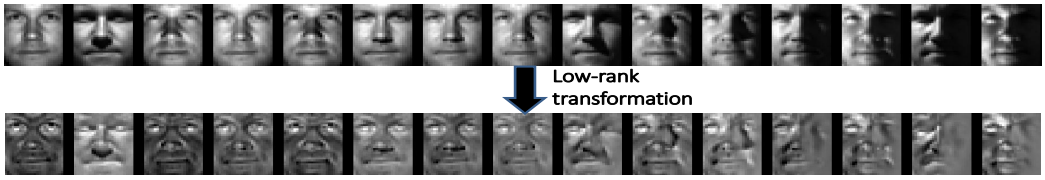
We report recognition accuracies in Table 5. We make the following observations. First, the recognition accuracy is increased from 91.77% to 99.10% by simply applying the learned transformation matrix to the original face images. Second, the best accuracy is obtained by first recovering the low-rank subspace for each subject, e.g., the third row in Figure 14a. Then, each transformed testing face, e.g., the second row in Figure 14b, is sparsely decomposed over the low-rank subspace of each subject through OMP, and classified to the subject with the minimal reconstruction error. A sparsity value 10 is used here for OMP. As shown in Figure 14c, the low-rank representation for each subject shows reduced variations caused by illumination. Third, the global transformation performs better here than class-based transformations, which can be due to the fact that illumination in this data set varies in a globally coordinated way across subjects. Last but not least, our method outperforms state-of-the-art sparse representation based face recognition methods.

5.5 Application to Face Recognition across Pose

We adopt the similar setup as described in Castillo and Jacobs (2009) to enable the comparison. In this experiment, we classify 68 subjects in three poses, frontal (c27), side (c05), and profile (c22), under lighting condition 12. We use the remaining poses as the training data.

For this example, we learn a class-based low-rank transformation matrix per subject from the training data. It is noted that the goal is to learn a transformation matrix to help in the classification, which may not necessarily correspond to the real geometric transform. Table 6 shows the face recognition accuracies under pose variations for the CMU PIE

(a) Low-rank decomposition of globally transformed training samples



(b) Globally transformed testing samples

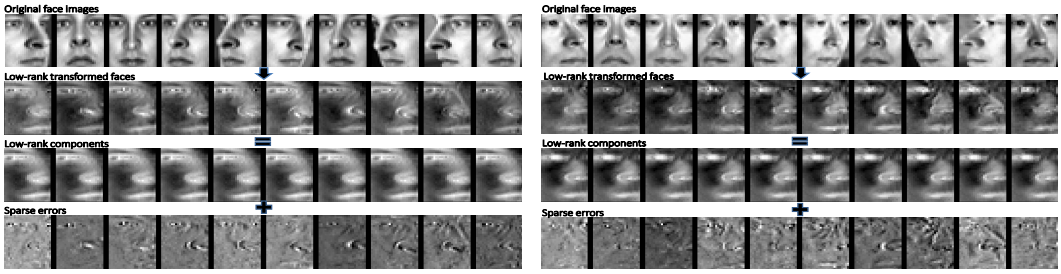
(c) Mean low-rank components for subjects in the training data

Figure 14: Face recognition across illumination using global low-rank transformation.

Method	Frontal (c27)	Side (c05)	Profile (c22)
SMD Castillo and Jacobs (2009)	83	82	57
Original+NN	39.85	37.65	17.06
Original(crop+flip)+NN	44.12	45.88	22.94
Class LRT+NN	98.97	96.91	67.65
Class LRT+OMP	**100**	**100**	**67.65**
Global LRT+NN	97.06	95.58	50
Global LRT+OMP	100	98.53	57.35

Table 6: Recognition accuracies (%) under pose variations for the CMU PIE data set.

data set (we applied the crop-and-flip step discussed in Figure 1.). We make the following observations. First, the recognition accuracy is dramatically increased after applying the learned transformations. Second, the best accuracy is obtained by recovering the low-rank subspace for each subject, e.g., the third row in Figure 15a and Figure 15b. Then, each transformed testing face, e.g., Figure 15c and Figure 15d, is sparsely decomposed over the low-rank subspace of each subject through OMP, and classified to the subject with the minimal reconstruction error, Section 4. Third, the class-based transformation performs better than the global transformation in this case. The choice between these two settings



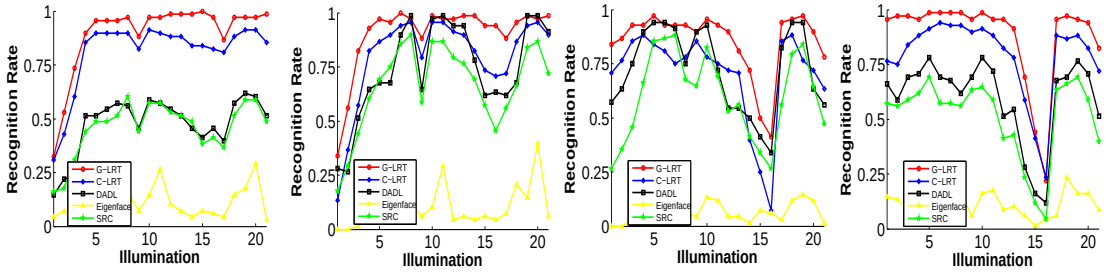
(a) Low-rank decomposition of class-based transformed training samples for *subject3*

(b) Low-rank decomposition of class-based transformed training samples for *subject1*

(c) class-based transformed testing samples for *subject3*

(d) class-based transformed testing samples for *subject1*

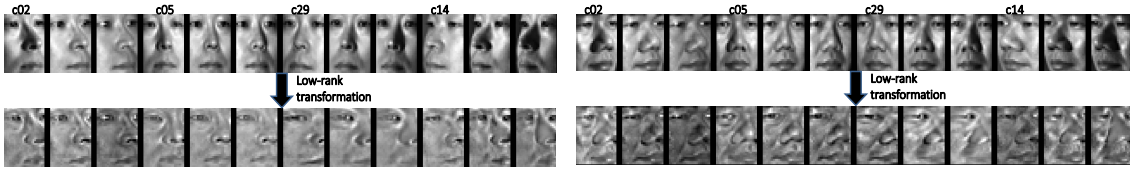
Figure 15: Face recognition across pose using class-based low-rank transformation. Note, for example in (c) and (d), how the learned transform reduces the pose-variability.



(a) Pose c02 (b) Pose c05 (c) Pose c29 (d) Pose c14

Figure 16: Face recognition accuracy under combined pose and illumination variations on the CMU PIE data set. The proposed methods are denoted as *G-LRT* in color red and *C-LRT* in color blue. The proposed methods significantly outperform the comparing methods, especially for extreme poses c02 and c14.

is data dependent. Last but not least, our method outperforms SMD, which the best of our knowledge, reported the best recognition performance in such experimental setup. However, SMD is an unsupervised method, and the proposed method requires training, still illustrating how a simple learned transform (note that applying it to the data at testing time if virtually free of cost), can significantly improve performance.



(a) Globally transformed testing samples for *subject1* (b) Globally transformed testing samples for *subject2*

Figure 17: Face recognition under combined pose and illumination variations using global low-rank transformation.

5.6 Application to Face Recognition across Illumination and Pose

To enable the comparison with Qiu et al. (Oct. 2012), we adopt their setup for face recognition under combined pose and illumination variations for the CMU PIE data set. We use 68 subjects in 5 poses, c22, c37, c27, c11 and c34, under 21 illumination conditions for training; and classify 68 subjects in 4 poses, c02, c05, c29 and c14, under 21 illumination conditions.

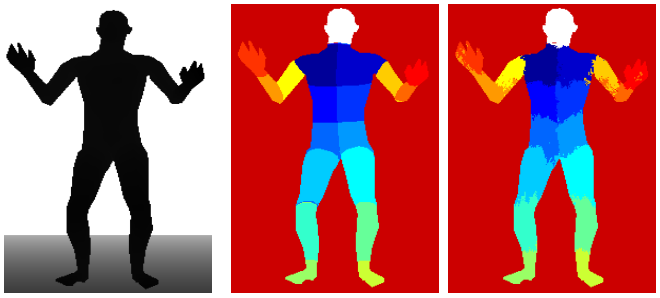
Three face recognition methods are adopted for comparisons: Eigenfaces Turk and Pentland (1991), SRC Wright et al. (2009), and DADL Qiu et al. (Oct. 2012). SRC and DADL are both state-of-the-art sparse representation methods for face recognition, and DADL adapts sparse dictionaries to the actual visual domains. As shown in Figure 16, the proposed methods, both the global LRT (G-LRT) and class-based LRT (C-LRT), significantly outperform the comparing methods, especially for extreme poses c02 and c14. Some testing examples using a global transformation are shown in Figure 17. We notice that the transformed faces for each subject exhibit reduced variations caused by pose and illumination.

5.7 Transformation Forest

In order to further illustrate the power of the framework here proposed, we briefly describe its use in combination with random forests, as discussed in detail in Qiu and Sapiro (2014). In this work we introduced a transformation-based learner model for random forest, further stressing how the proposed transformation learning can be combined with other successful classification techniques beyond subspace techniques. The weak learner at each split node plays a crucial role in a classification tree. We optimized the splitting by learning a two-class transformation $\mathbf{T}$ at each split node, and observed significantly performance improvements in various real-world applications, such as scene classification and 3D pose estimation (Figure 18). In particular, we experimentally demonstrated how learning such transform at each node reduces by 1-2 orders of magnitude the number of trees in the random forest.

5.8 Discussion on the Size of the Transformation Matrix T

In the experiments presented above, we learned a square linear transformation. For example, if images are resized to 16×16, the learned subspace transformation $\mathbf{T}$ is of size 256×256. If we learn a transformation of size $r \times 256$ with $r < 256$, we enable dimension reduction while performing subspace transformation (feature learning). Through experiments, we



(a) Depth. (b) Ground truth. (c) Prediction.

Figure 18: Body parts prediction from a depth image using transformation forests (Qiu and Sapiro, 2014). With the learned transform we classify 20 regions (19 body parts and one background) with 55.5% correct for a single tree (only about 40% with standard trees), and achieve already 73.12% with just 30 trees (hundreds are normally used with standard trees).

notice that the peak clustering accuracy is usually obtained when r is smaller than the dimension of the ambient space. For example, in Figure 13, through exhaustive search for the optimal r, we observe the misclassification rate reduced from 2.38% to 0% for subjects $\{2, 3\}$ at $r = 96$, and from 4.23% to 0% for subjects $\{4, 5, 6\}$ at $r = 40$. As discussed before, this provides a framework to sense for clustering and classification, connecting the work presented here with the extensive literature on compressed sensing, and in particular for sensing design, e.g., Carson et al. (2012). We plan to study in detail the optimal size of the learned transformation matrix for subspace clustering and classification, including its potential connection with the number of subspaces in the data, and further investigate such connections with compressive sensing.

6. Conclusion

We introduced a subspace low-rank transformation approach for subspace clustering and classification. Using nuclear norm as the optimization criteria, we learn a subspace transformation that reduces variations within the subspaces, and increases separations between the subspaces. We demonstrated that the proposed approach significantly outperforms state-of-the-art methods for subspace clustering and classification, and provided some theoretical support to these experimental results.

Numerous venues of research are opened by the framework introduced here. At the theoretical level, extending the analysis to the noisy case is needed. Furthermore, understanding the virtues of the global vs the class-dependent transform is both important and interesting, as it is the study of the framework in its compressed dimensionality form. Beyond this, considering the proposed approach as a feature extraction technique, its combination with other successful clustering and classification techniques is the subject of current research.

Acknowledgments

Work partially supported by ONR, NGA, ARO, AFOSR (NSSEFF), and NSF. We thank Dr. Pablo Sprechmann, Dr. Ehsan Elhamifar, Ching-Hui Chen, and Dr. Mariano Tepper for important feedback on this work. The AE and reviewers did an outstanding job in helping us improve this paper.

Appendix A. Proof of Theorem 1

Proof:

$$||\mathbf{A}||_* + ||\mathbf{B}||_* = |||[\mathbf{A}\ \mathbf{0}]|||_* + |||[\mathbf{0}\ \mathbf{B}]|||_* \geq |||[\mathbf{A}\ \mathbf{0}] + [\mathbf{0}\ \mathbf{B}]|||_* = |||[\mathbf{A}, \mathbf{B}]|||_*$$

■

Appendix B. Proof of Theorem 2

Proof: We perform the singular value decomposition of $\mathbf{A}$ and $\mathbf{B}$ as

$$\mathbf{A} = [\mathbf{U_{A1}}\mathbf{U_{A2}}] \begin{bmatrix} \mathbf{\Sigma_A} & 0 \\ 0 & 0 \end{bmatrix} [\mathbf{V_{A1}}\mathbf{V_{A2}}]', \quad \mathbf{B} = [\mathbf{U_{B1}}\mathbf{U_{B2}}] \begin{bmatrix} \mathbf{\Sigma_B} & 0 \\ 0 & 0 \end{bmatrix} [\mathbf{V_{B1}}\mathbf{V_{B2}}]',$$

where the diagonal entries of $\mathbf{\Sigma_A}$ and $\mathbf{\Sigma_B}$ contain non-zero singular values. We have

$$\mathbf{AA'} = [\mathbf{U_{A1}}\mathbf{U_{A2}}] \begin{bmatrix} \mathbf{\Sigma_A}^2 & 0 \\ 0 & 0 \end{bmatrix} [\mathbf{U_{A1}}\mathbf{U_{A2}}]', \quad \mathbf{BB'} = [\mathbf{U_{B1}}\mathbf{U_{B2}}] \begin{bmatrix} \mathbf{\Sigma_B}^2 & 0 \\ 0 & 0 \end{bmatrix} [\mathbf{U_{B1}}\mathbf{U_{B2}}]'.$$

The column spaces of $\mathbf{A}$ and $\mathbf{B}$ are considered to be orthogonal, i.e., $\mathbf{U_{A1}}'\mathbf{U_{B1}} = 0$. The above can be written as

$$\mathbf{AA'} = [\mathbf{U_{A1}}\mathbf{U_{B1}}] \begin{bmatrix} \mathbf{\Sigma_A}^2 & 0 \\ 0 & 0 \end{bmatrix} [\mathbf{U_{A1}}\mathbf{U_{B1}}]', \quad \mathbf{BB'} = [\mathbf{U_{A1}}\mathbf{U_{B1}}] \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{\Sigma_B}^2 \end{bmatrix} [\mathbf{U_{A1}}\mathbf{U_{B1}}]'.$$

Then, we have

$$[\mathbf{A}, \mathbf{B}][\mathbf{A}, \mathbf{B}]' = \mathbf{AA'} + \mathbf{BB'} = [\mathbf{U_{A1}}\mathbf{U_{B1}}] \begin{bmatrix} \mathbf{\Sigma_A}^2 & 0 \\ 0 & \mathbf{\Sigma_B}^2 \end{bmatrix} [\mathbf{U_{A1}}\mathbf{U_{B1}}]'.$$

The nuclear norm $||\mathbf{A}||_*$ is the sum of the square root of the singular values of $\mathbf{AA'}$. Thus, $|||[\mathbf{A}, \mathbf{B}]|||_* = ||\mathbf{A}||_* + ||\mathbf{B}||_*$. ■

Appendix C. The Concave-Convex Procedure

We use a simple projected subgradient method to search for the transformation matrix $\mathbf{T}$ that minimizes (6). Before describing it, we should note that the problem is non-differentiable and non-convex, and it deserves in its own right a proper study of efficient optimization techniques, which is of course not the focus of this paper. The development of more advanced optimization techniques will further improve the performance of the proposed framework. We selected a simple subgradient-based approach since the goal of this

paper is to present the framework, and already this simple optimization leads to very fast convergence and excellent performance as detailed in Section 5, with significant improvements in performance when compared to prior state-of-the-art.

The objective function (6) is a D.C. (difference of convex functions) program, and the concave-convex procedure (CCCP) is a majorization-minimization algorithm often adopted to solve D.C. programs as a sequence of convex programs (Yuille and Rangarajan, 2003; Sriperumbudur and Lanckriet, 2012; Dinh and An, 1997). CCCP is used in many machine learning algorithms such as transductive SVMs (Collobert et al., 2006), sparse PCA (Sriperumbudur et al., 2007), and SVM feature selection (Neumann et al., 2005).

Initialize $\mathbf{T}^{(0)}$ with the identity matrix ;
repeat

$$\begin{aligned}\mathbf{T}^{(t+1)} =& \arg\min_{\mathbf{T}} \mathcal{J}_{vex}(\mathbf{T}) + \partial\mathcal{J}_{cav}(\mathbf{T}^{(t)})\mathbf{T} \\ =& \arg\min_{\mathbf{T}} \sum_{c=1}^{C} ||\mathbf{T}\mathbf{Y}_c||_* - \text{trace}(\partial||\mathbf{T}^{(t)}\mathbf{Y}||_*\mathbf{Y}'\mathbf{T}').\end{aligned} \tag{17}$$

until *convergence or stopping criteria*;

Algorithm 2: The Concave-Convex Procedure (CCCP).

Input: An $m \times n$ matrix $\mathbf{A}$, a small threshold value δ
Output: A subgradient of the nuclear norm $\partial||\mathbf{A}||_*$.
begin

1. Perform singular value decomposition:
$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}$;
2. $s \leftarrow$ the number of singular values smaller than δ ,
3. Partition $\mathbf{U}$ and $\mathbf{V}$ as
$\mathbf{U} = [\mathbf{U}^{(1)}, \mathbf{U}^{(2)}]$, $\mathbf{V} = [\mathbf{V}^{(1)}, \mathbf{V}^{(2)}]$;
where $\mathbf{U}^{(1)}$ and $\mathbf{V}^{(1)}$ have $(n-s)$ columns.
4. Generate a random matrix $\mathbf{B}$ of the size $(m-n+s) \times s$,
$\mathbf{B} \leftarrow \frac{\mathbf{B}}{||\mathbf{B}||}$;
5. $\partial||\mathbf{A}||_* \leftarrow \mathbf{U}^{(1)}\mathbf{V}^{(1)'} + \mathbf{U}^{(2)}\mathbf{B}\mathbf{V}^{(2)'}$;
6. Return $\partial||\mathbf{A}||_*$;

end

Algorithm 3: An approach to evaluate a subgradient of matrix nuclear norm.

Our D.C. cost function $\mathcal{J}(\mathbf{T})$ can be rewritten as the sum of a convex part $\mathcal{J}_{vex}(\mathbf{T})$ and a concave part $\mathcal{J}_{cav}(\mathbf{T})$, i.e.,

$$\begin{aligned}\mathcal{J}(\mathbf{T}) =& \mathcal{J}_{vex}(\mathbf{T}) + \mathcal{J}_{cav}(\mathbf{T}) \\ =& [\sum_{c=1}^{C} ||\mathbf{T}\mathbf{Y}_c||_*] + [-||\mathbf{T}\mathbf{Y}||_*].\end{aligned}$$

In each iteration of the CCCP procedure, Algorithm 2, we approximate the concave part using its subgradient $\partial\mathcal{J}_{cav}$, and minimize the resulting convex sub-problem. Note that the

first term in (17) is the convex term in (6), and the added second term is a linear term on $\mathbf{T}$ using a subgradient of the concave term in (6) evaluated at the current iteration. $\partial||\cdot||_*$ is a subgradient of the nuclear norm $||\cdot||_*$, which can be evaluated using the simple approach shown in Algorithm 3 (Watson, 1992). Formal convergence analysis of CCCP for differentiable cases can be found in Yuille and Rangarajan (2003) and Sriperumbudur and Lanckriet (2012). Though the objective function (6) is non-differentiable, we still observe empirical convergence in all experiments, see Figure8 and Figure11n.

We provide here more details about Algorithm 2. During each CCCP iteration, we solve the convex sub-objective (17) using the subgradient method, i.e., using a constant step size ν ($\nu > 0$), we iteratively take a step in the negative direction of subgradient, and the subgradient is evaluated as

$$\sum_{c=1}^{C} \partial||\mathbf{T}\mathbf{Y}_c||_*\mathbf{Y}_c' - \partial||\mathbf{T}^{(\mathbf{t})}\mathbf{Y}||_*\mathbf{Y}'. \tag{18}$$

Using a constant step size, the subgradient method is guaranteed to converge to within some range of the optimal value for a convex problem (convergence to the optimal value is guaranteed by using a diminishing step size with an infinite travel condition) (Boyd et al., 2003). Therefore, given $\mathbf{T}^{(t+1)}$ as the minimizer found for the convex sub-problem (17) using the subgradient method, we have for (17),

$$\begin{aligned}\sum_{c=1}^{C} ||\mathbf{T}^{(t+1)}\mathbf{Y}_c||_* - \text{trace}(\partial||\mathbf{T}^{(t)}\mathbf{Y}||_*\mathbf{Y}'\mathbf{T}^{(t+1)'}) \\ \leq \sum_{c=1}^{C} ||\mathbf{T}^{(t)}\mathbf{Y}_c||_* - \text{trace}(\partial||\mathbf{T}^{(t)}\mathbf{Y}||_*\mathbf{Y}'\mathbf{T}^{(t)'}),\end{aligned} \tag{19}$$

and from the concavity of the second term in (6), we have

$$-||\mathbf{T}^{(t+1)}\mathbf{Y}||_* \leq -||\mathbf{T}^{(t)}\mathbf{Y}||_* - \text{trace}(\partial||\mathbf{T}^{(t)}\mathbf{Y}||_*\mathbf{Y}'(\mathbf{T}^{(t+1)} - \mathbf{T}^{(t)})'). \tag{20}$$

By summing (19) and (20), we obtain

$$\sum_{c=1}^{C} ||\mathbf{T}^{(t+1)}\mathbf{Y}_c||_* - ||\mathbf{T}^{(t+1)}\mathbf{Y}||_* \leq \sum_{c=1}^{C} ||\mathbf{T}^{(t)}\mathbf{Y}_c||_* - ||\mathbf{T}^{(t)}\mathbf{Y}||_*. \tag{21}$$

Thus, the objective (6) is non-increasing after each CCCP iteration, and is bounded from below by 0 (shown in Section 2) for our non-differentiable case. For efficiency considerations, while solving the convex sub-objective function (17), we perform only one iteration of the subgradient method to obtain a simplified method, and still observe empirical convergence in all experiments, as shown in Figure8 and Figure11n.

The norm constraint $||\mathbf{T}||_2 = 1$ is adopted in our formulation to prevent the trivial solution $\mathbf{T} = 0$. By initializing $\mathbf{T}^{(0)}$ with the identity matrix, we observed no trivial solution convergence in all experiments, such as the normalization free case in Figure8.

As shown in Douglas et al. (2000), the norm constraint $||\mathbf{T}||_2 = 1$ can be incorporated to a gradient-based algorithm using various alternatives, e.g., Lagrange multipliers, coefficient

normalization, and gradients in the tangent space. We implement the coefficient normalization method, i.e., after obtaining $\mathbf{T}^{(t+1)}$ from (17), we normalize $\mathbf{T}^{(t+1)}$ via $\frac{\mathbf{T}^{(t+1)}}{||\mathbf{T}^{(t+1)}||}$. In other words, we normalize the length of $\mathbf{T}^{(t+1)}$ without changing its direction. As discussed in Douglas et al. (2000), the problem of minimizing a cost function subject to a norm constraint forms the basis for many important tasks, and gradient-based algorithms are often used along with the norm constraint. Though it is expected that a norm constraint does not change the convergence behavior of a gradient algorithm (Douglas et al., 2000; Fuhrmann and Liu, 1984), Figure8, to the best of our knowledge, a formal analysis of these issues is still missing.

References

R. Basri and D. W. Jacobs. Lambertian reflectance and linear subspaces. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 25(2):218–233, 2003.

M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15:1373–1396, 2003.

S. Boyd, L. Xiao, and A. Mutapcic. Subgradient method. *Notes for EE392o, Stanford University*, 2003.

E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *J. ACM*, 58(3):11:1–11:37, June 2011.

W. R. Carson, M. Chen, M. R. D. Rodrigues, R. Calderbank, and L. Carin. Communications-inspired projection design with application to compressive sensing. *SIAM J. Imaging Sci.*, 5(4):1185–1212, 2012.

C. Castillo and D. Jacobs. Using stereo matching for 2-D face recognition across pose. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 31:2298–2304, 2009.

G. Chen and G. Lerman. Spectral curvature clustering (SCC). *International Journal of Computer Vision*, 81(3):317–330, 2009.

R. Collobert, F. Sinz, J. Weston, and L. Bottou. Large scale transductive svms. *J. Mach. Learn. Res.*, 7:1687–1712, December 2006.

T. P. Dinh and L. T. H. An. Convex analysis approach to d.c. programming: Theory, algorithms and applications. *Acta Mathematica Vietnamica*, 22(1):289355, 1997.

S. C. Douglas, S. Amari, and S. Y. Kung. On gradient adaptation with unit-norm constraints. *IEEE Trans. on Signal Processing*, 48(6):1843–1847, 2000.

E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 2013. To appear.

M. Fazel. *Matrix Rank Minimization with Applications*. PhD thesis, Stanford University, 2002.

D. R. Fuhrmann and B. Liu. An iterative algorithm for locating the minimal eigenvector of a symmetric matrix. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Dallas, TX, 1984.

A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 23(6):643–660, June 2001.

A. Goh and R. Vidal. Segmenting motions of different types by unsupervised manifold clustering. In *Proc. IEEE Computer Society Conf. on Computer Vision and Patt. Recn.*, Minneapolis, Minnesota, 2007.

T. Hastie and P. Y. Simard. Metrics and models for handwritten character recognition. *Statistical Science*, 13(1):54–65, 1998.

Z. Jiang, Z. Lin, and L. S. Davis. Learning a discriminative dictionary for sparse coding via label consistent K-SVD. In *Proc. IEEE Computer Society Conf. on Computer Vision and Patt. Recn.*, Colorado springs, CO, 2011.

O. Kuybeda, G. A. Frank, A. Bartesaghi, M. Borgnia, S. Subramaniam, and G. Sapiro. A collaborative framework for 3D alignment and classification of heterogeneous subvolumes in cryo-electron tomography. *Journal of Structural Biology*, 181:116–127, 2013.

G. Liu, Z. Lin, and Y. Yu. Robust subspace segmentation by low-rank representation. In *International Conference on Machine Learning*, Haifa, Israel, 2010.

U. Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, December 2007.

Y. Ma, H. Derksen, W. Hong, and J. Wright. Segmentation of multivariate mixed data via lossy data coding and compression. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 29 (9):1546–1562, 2007.

G. Marsaglia and G. P. H. Styan. When does rank (a + b) = rank(a)+rank(b)? *Canad. Math. Bull.*, 15(3), 1972.

J. Miao and A. Ben-Israel. On principal angles between subspaces in R_n. *Linear Algebra and its Applications*, 171(0):81 – 98, 1992.

J. Neumann, C. Schnörr, and G. Steidl. Combined SVM-based feature selection and classification. *Mach. Learn.*, 61(1-3):129–150, November 2005.

Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. *Proc. 27th Asilomar Conference on Signals, Systems and Computers*, pages 40–44, Nov. 1993.

Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma. RASL: Robust alignment by sparse and low-rank decomposition for linearly correlated images. In *Proc. IEEE Computer Society Conf. on Computer Vision and Patt. Recn.*, San Francisco, USA, 2010.

Q. Qiu and G. Sapiro. Learning transformations for classification forests. In *International Conference on Learning Representations*, Banff, Canada, 2014.

Q. Qiu, V. Patel, P. Turaga, and R. Chellappa. Domain adaptive dictionary learning. In *Proc. European Conference on Computer Vision*, Florence, Italy, Oct. 2012.

B. Recht, M. Fazel, and P. A. Parrilo. Guaranteed minimum rank solutions to linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3):471–501, 2010.

S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.

L. K. Saul and S. T. Roweis. An introduction to locally linear embedding. 2000. URL `http://www.cs.nyu.edu/~roweis/lle/publications.html`.

X. Shen and Y. Wu. A unified approach to salient object detection via low rank matrix recovery. In *Proc. IEEE Computer Society Conf. on Computer Vision and Patt. Recn.*, Rhode Island, USA, 2012.

T. Sim, S. Baker, and M. Bsat. The CMU pose, illumination, and expression (PIE) database. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 25(12):1615 –1618, Dec. 2003.

M. Soltanolkotabi and E. J. Candes. A geometric analysis of subspace clustering with outliers. *The Annals of Statistics*, 40(4):2195–2238, 2012.

M. Soltanolkotabi, E. Elhamifar, and E. J. Candès. Robust subspace clustering. *CoRR*, abs/1301.2603, 2013. URL `http://arxiv.org/abs/1301.2603`.

P. Sprechmann, A. M. Bronstein, and G. Sapiro. Learning efficient sparse and low rank models. *CoRR*, abs/1212.3631, 2012. URL `http://arxiv.org/abs/1212.3631`.

B. K. Sriperumbudur and G. R. G. Lanckriet. A proof of convergence of the concave-convex procedure using zangwill's theory. *Neural Computation*, 24(6):1391–1407, 2012.

B. K. Sriperumbudur, D. A. Torres, and G. R. G. Lanckriet. Sparse eigen methods by d.c. programming. In *International Conference on Machine Learning*, 2007.

C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9:137–154, 1992.

M.A. Turk and A.P. Pentland. Face recognition using eigenfaces. In *Proc. IEEE Computer Society Conf. on Computer Vision and Patt. Recn.*, Maui, Hawaii, 1991.

R. Vidal. Subspace clustering. *Signal Processing Magazine, IEEE*, 28(2):52–68, 2011.

R. Vidal, Yi Ma, and S. Sastry. Generalized principal component analysis (GPCA). In *Proc. IEEE Computer Society Conf. on Computer Vision and Patt. Recn.*, Madison, Wisconsin, 2003.

J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *Proc. IEEE Computer Society Conf. on Computer Vision and Patt. Recn.*, San Francisco, USA, 2010.

Y. Wang and H. Xu. Noisy sparse subspace clustering. In *International Conference on Machine Learning*, Atlanta, USA, 2013.

G. A. Watson. Characterization of the subdifferential of some matrix norms. *Linear Algebra and Applications*, 170:1039–1053, 1992.

J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 31(2):210–227, 2009.

J. Yan and M. Pollefeys. A general framework for motion segmentation: independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *Proc. European Conference on Computer Vision*, Graz, Austria, 2006.

A. L. Yuille and A. Rangarajan. The concave-convex procedure. *Neural Computation*, 4: 915–936, 2003.

Q. Zhang and B. Li. Discriminative k-SVD for dictionary learning in face recognition. In *Proc. IEEE Computer Society Conf. on Computer Vision and Patt. Recn.*, San Francisco, CA, 2010.

T. Zhang, A. Szlam, Y. Wang, and G. Lerman. Hybrid linear modeling via local best-fit flats. *International Journal of Computer Vision*, 100(3):217–240, 2012.

Z. Zhang, X. Liang, A. Ganesh, and Y. Ma. TILT: transform invariant low-rank textures. In *Proc. Asian conference on Computer vision*, Queenstown, New Zealand, 2011.

X. Zhu and D. Ramanan. Face detection, pose estimation and landmark localization in the wild. In *Proc. IEEE Computer Society Conf. on Computer Vision and Patt. Recn.*, Providence, Rhode Island, 2012.

Journal of Machine Learning Research 16 (2015) 227-254 Submitted 12/13; Revised 7/14; Published 2/15

Multi-layered Gesture Recognition with Kinect

Feng Jiang FJIANG@HIT.EDU.CN
School of Computer Science and Technology
Harbin Institute of Technology, Harbin 150001, China

Shengping Zhang S.ZHANG@HIT.EDU.CN
School of Computer Science and Technology
Harbin Institute of Technology, Weihai 264209, China

Shen Wu WU.SHEN.ELTSHAN@GMAIL.COM
Yang Gao LAMBYY.HIT@GMAIL.COM
Debin Zhao DBZHAO@HIT.EDU.CN
School of Computer Science and Technology
Harbin Institute of Technology, Harbin 150001, China

Editors: Isabelle Guyon, Vassilis Athitsos, and Sergio Escalera

Abstract

This paper proposes a novel multi-layered gesture recognition method with Kinect. We explore the essential linguistic characters of gestures: the components concurrent character and the sequential organization character, in a multi-layered framework, which extracts features from both the segmented semantic units and the whole gesture sequence and then sequentially classifies the motion, location and shape components. In the first layer, an improved principle motion is applied to model the motion component. In the second layer, a particle-based descriptor and a weighted dynamic time warping are proposed for the location component classification. In the last layer, the spatial path warping is further proposed to classify the shape component represented by unclosed shape context. The proposed method can obtain relatively high performance for one-shot learning gesture recognition on the ChaLearn Gesture Dataset comprising more than 50, 000 gesture sequences recorded with Kinect.

Keywords: gesture recognition, Kinect, linguistic characters, multi-layered classification, principle motion, dynamic time warping

1. Introduction

Gestures, an unsaid body language, play very important roles in daily communication. They are considered as the most natural means of communication between humans and computers (Mitra and Acharya, 2007). For the purpose of improving humans' interaction with computers, considerable work has been undertaken on gesture recognition, which has wide applications including sign language recognition (Vogler and Metaxas, 1999; Cooper et al., 2012), socially assistive robotics (Baklouti et al., 2008), directional indication through pointing (Nickel and Stiefelhagen, 2007) and so on (Wachs et al., 2011).

Based on the devices used to capture gestures, gesture recognition can be roughly categorized into two groups: wearable sensor-based methods and optical camera-based methods. The representative device in the first group is the data glove (Fang et al., 2004), which is

capable of exactly capturing the motion parameters of the user's hands and therefore can achieve high recognition performance. However, these devices affect the naturalness of the user interaction. In addition, they are also expensive, which restricts their practical applications (Cooper et al., 2011). Different from the wearable devices, the second group of devices are optical cameras, which record a set of images overtime to capture gesture movements in a distance. The gesture recognition methods based on these devices recognize gestures by analyzing visual information extracted from the captured images. That is why they are also called vision-based methods. Although optical cameras are easy to use and also inexpensive, the quality of the captured images is sensitive to lighting conditions and cluttered backgrounds, thus it is very difficult to detect and track the hands robustly, which largely affects the gesture recognition performance.

Recently, the Kinect developed by Microsoft was widely used in both industry and research communities (Shotton et al., 2011). It can capture both RGB and depth images of gestures. With depth information, it is not difficult to detect and track the user's body robustly even in noisy and cluttered backgrounds. Due to the appealing performance and also reasonable cost, it has been widely used in several vision tasks such as face tracking (Cai et al., 2010), hand tracking (Oikonomidis et al., 2011), human action recognition (Wang et al., 2012) and gesture recognition (Doliotis et al., 2011; Ren et al., 2013). For example, one of the earliest methods for gesture recognition using Kinect is proposed in Doliotis et al. (2011), which first detects the hands using scene depth information and then employs Dynamic Time Warping for recognizing gestures. Ren et al. (2013) extracts the static finger shape features from depth images and measures the dissimilarity between shape features for classification. Although, Kinect facilitates us to detect and track the hands, exact segmentation of finger shapes is still very challenging since the fingers are very small and form many complex articulations.

Although postures and gestures are frequently considered as being identical, there are significant differences (Corradini, 2002). A posture is a static pose, such as making a palm posture and holding it in a certain position, while a gesture is a dynamic process consisting of a sequence of the changing postures over a short duration. Compared to postures, gestures contain much richer motion information, which is important for distinguishing different gestures especially those ambiguous ones. The main challenge of gesture recognition lies in the understanding of the unique characters of gestures. Exploring and utilizing these characters in gesture recognition are crucial for achieving desired performance. Two crucial linguistic models of gestures are the phonological model drawn from the component concurrent character (Stokoe, 1960) and the movement-hold model drawn from the sequential organization character (Liddell and Johnson, 1989). The component concurrent character indicates that complementary components, namely motion, location and shape components, simultaneously characterize a unique gesture. Therefore, an ideal gesture recognition method should have the ability of capturing, representing and recognizing these simultaneous components. On the other hand, the movement phases, i.e., the transition phases, are defined as periods during which some components, such as the shape component, are in transition; while the holding phases are defined as periods during which all components are static. The sequential organization character characterizes a gesture as a sequential arrangement of movement phases and holding phases. Both the movement phases and the holding phases are defined as semantic units. Instead of taking the entire gesture sequence as input, the movement-

hold model inspires us to segment a gesture sequence into sequential semantic units and then extract specific features from them. For example, for the frames in a holding phase, shape information is more discriminative for classifying different gestures.

It should be noted that the component concurrent character and the sequential organization character demonstrate the essences of gestures from spatial and temporal aspects, respectively. The former indicates which kinds of features should be extracted. The later implies that utilizing the cycle of movement and hold phases in a gesture sequence can accurately represent and model the gesture. Considering these two complementary characters together provides us a way to improve gesture recognition. Therefore, we developed a multi-layered classification framework for gesture recognition. The architecture of the proposed framework is shown in Figure 1, which contains three layers: the motion component classifier, the location component classifier, and the shape component classifier. Each of the three layers analyzes its corresponding component. The output of one layer limits the possible classification in the next layer and these classifiers complement each other for the final gesture classification. Such a multi-layered architecture assures achieving high recognition performance while being computationally inexpensive.

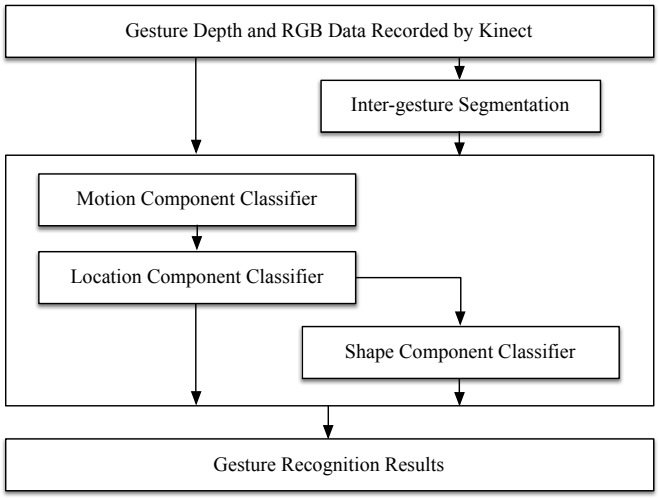


Figure 1: Multi-layered gesture recognition architecture.

The main contributions of this paper are summarized as follows:

- The phonological model (Stokoe, 1960) of gestures inspires us to propose a novel multi-layered gesture recognition framework, which sequentially classifies the motion, location and shape components and therefore achieves higher recognition accuracy while having low computational complexity.
- Inspired by the linguistic sequential organization of gestures (Liddell and Johnson, 1989), the matching process between two gesture sequences is divided into two steps: their semantic units are matched first, and then the frames inside the semantic units are further registered. A novel particle-based descriptor and a weighted dynamic time warping are proposed to classify the location component.

- The spatial path warping is proposed to classify the shape component represented by unclosed shape context, which is improved from the original shape context but the computation complexity is reduced from $O(n^3)$ to $O(n^2)$.

Our proposed method participated the one-shot learning ChaLearn gesture challenge and was top ranked (Guyon et al., 2013). The ChaLearn Gesture Dataset (CGD 2011) (Guyon et al., 2014) is designed for one-shot learning and comprises more than 50, 000 gesture sequences recorded with Kinect. The remainder of the paper is organized as follows. Related work is reviewed in Section 2. The detailed descriptions of the proposed method are presented in Section 3. Extensive experimental results are reported in Section 4. Section 5 concludes the paper.

2. Related Work

Vision based gesture recognition methods encompasses two main categories: three dimensional (3D) model based methods and appearance based methods. The former computes a geometrical representation using the joint angles of a 3D articulated structure recovered from a gesture sequence, which provides a rich description that permits a wide range of gestures. However, computing a 3D model has high computational complexity (Oikonomidis et al., 2011). In contrast, appearance based methods extract appearance features from a gesture sequence and then construct a classifier to recognize different gestures, which have been widely used in vision based gesture recognition (Dardas, 2012). The proposed multi-layered gesture recognition falls into the appearance based methods.

2.1 Feature Extraction and Classification

The well known features used for gesture recognition are color (Awad et al., 2006; Maraqa and Abu-Zaiter, 2008), shapes (Ramamoorthy et al., 2003; Ong and Bowden, 2004) and motion (Cutler and Turk, 1998; Mahbub et al., 2013). In early work, color information is widely used to segment the hands of a user. To simplify the color based segmentation, the user is required to wear single or differently colored gloves (Kadir et al., 2004; Zhang et al., 2004). The skin color models are also used (Stergiopoulou and Papamarkos, 2009; Maung, 2009) where a typical restriction is wearing of long sleeved clothes. When it is difficult to exploit color information to segment the hands from an image (Wan et al., 2012b), motion information extracted from two consecutive frames is used for gesture recognition. Agrawal and Chaudhuri (2003) explores the correspondences between patches in adjacent frames and uses 2D motion histogram to model the motion information. Shao and Ji (2009) computes optical flow from each frame and then uses different combinations of the magnitude and direction of optical flow to compute a motion histogram. Zahedi et al. (2005) combines skin color features and different first- and second-order derivative features to recognize sign language. Wong et al. (2007) uses PCA on motion gradient images of a sequence to obtain features for a Bayesian classifier. To extract motion features, Cooper et al. (2011) extends Haar-like features from spatial domain to spatio-temporal domain and proposes volumetric Haar-like features.

The features introduced above are usually extracted from RGB images captured by a traditional optical camera. Due to the nature of optical sensing, the quality of the captured

images is sensitive to lighting conditions and cluttered backgrounds, thus the extracted features from RGB images are not robust. In contrast, depth information from a calibrated camera pair (Rauschert et al., 2002) or direct depth sensors such as LiDAR (Light Detection and Ranging) is more robust to noises and illumination changes. More importantly, depth information is useful for discovering the distance between the hands and body orthogonal to the image plane, which is an important cue for distinguishing some ambiguous gestures. Because the direct depth sensors are expensive, inexpensive depth cameras, e.g., Microsoft's Kinect, have been recently used in gesture recognition (Ershaed et al., 2011; Wu et al., 2012b). Although the skeleton information offered by Kinect is more effective in the expression of human actions than pure depth data, there are some cases that skeleton cannot be extracted correctly, such as interaction between human body and other objects. Actually, in the CHALERAN gesture challenge (Guyon et al., 2013), the skeleton information is not allowed to use. To extract more robust features from Kinect depth images for gesture recognition, Ren et al. (2013) proposes the part based finger shape features, which do not depend on the accurate segmentation of the hands. Wan et al. (2013, 2014b) extend SIFT to spatio-temporal domain and propose 3D EMoSIFT and 3D SMoSIFT to extract features from RGB and depth images, which are invariant to scale and rotation, and have more compact and richer visual representations. Wan et al. (2014a) proposes a discriminative dictionary learning method on 3D EMoSIFT features based on mutual information and then uses sparse reconstruction for classification. Based on 3D Histogram of Flow (3DHOF) and Global Histogram of Oriented Gradient (GHOG), Fanello et al. (2013) applies adaptive sparse coding to capture high-level feature patterns. Wu et al. (2012a) utilizes both RGB and depth information from Kinect and an extended-MHI representation is adopted as the motion descriptors.

The performance of a gesture recognition method is not only related to the used features but also to the adopted classifiers. Many classifiers can be used for gesture recognition, e.g., Dynamic Time Warping (DTW) (Reyes et al., 2011; Lichtenauer et al., 2008; Sabinas et al., 2013), linear SVMs (Fanello et al., 2013), neuro-fuzzy inference system networks (Al-Jarrah and Halawani, 2001), hyper rectangular composite NNs (Su, 2000), and 3D Hopfield NN (Huang and Huang, 1998). Due to the ability of modeling temporal signals, Hidden Markov Model (HMM) is possibly the most well known classifier for gesture recognition. Bauer (Bauer and Kraiss, 2002) proposes a 2D motion model and performs gesture recognition with HMM. Vogler (2003) presents a parallel HMM algorithm to model gestures, which can recognize continuous gestures. Fang et al. (2004) proposes a self-organizing feature maps/hidden Markov model (SOFM/HMM) for gesture recognition in which SOFM is used as an implicit feature extractor for continuous HMM. Recently, Wan et al. (2012a) proposes ScHMM to deal with the gesture recognition where sparse coding is adopted to find succinct representations and Lagrange dual is applied to obtain a codebook.

2.2 One-shot Learning Gesture Recognition and Gesture Characters

Although a large number of work has been done, gesture recognition is still very challenging and has been attracting increasing interests. One motivation is to overcome the well-known overfitting problem when training samples are insufficient. The other one is to further improve gesture recognition by developing novel features and classifiers.

In the case of training samples being insufficient, most of classification methods are very likely to overfit. Therefore, developing gesture recognition methods that use only a small training data set is necessary. An extreme example is the one-shot learning that uses only one training sample per class for training. The proposed work in this paper is also for one-shot learning. In the literature, several previous work has been focused on one-shot learning. In Lui (2012a), gesture sequences are viewed as third-order tensors and decomposed to three Stiefel Manifolds and a natural metric is inherited from the factor manifolds. A geometric framework for least square regression is further presented and applied to gesture recognition. Mahbub et al. (2013) proposes a space-time descriptor and applies Motion History Imaging (MHI) techniques to track the motion flow in consecutive frames. The Euclidean distance based classifiers is used for gesture recognition. Seo and Milanfar (2011) presents a novel action recognition method based on space-time locally adaptive regression kernels and the matrix cosine similarity measure. Malgireddy et al. (2012) presents an end-to-end temporal Bayesian framework for activity classification. A probabilistic dynamic signature is created for each activity class and activity recognition becomes a problem of finding the most likely distribution to generate the test video. Escalante et al. (2013) introduces principal motion components for one-shot learning gesture recognition. 2D maps of motion energy are obtained per each pair of consecutive frames in a video. Motion maps associated to a video are further processed to obtain a PCA model, which is used for gesture recognition with a reconstruction-error approach. More one-shot learning gesture recognition methods are summarized by Guyon et al. (2013).

The intrinsic difference between gesture recognition and other recognition problems is that gesture communication is highly complex and owns its unique characters. Therefore, it is crucial to develop specified features and classifiers for gesture recognition by exploring the unique characters of gestures as explained in Section 1. There are some efforts toward this direction and some work has modeled the component concurrent or sequential organization and achieved significant progress. To capture meaningful linguistic components of gestures, Vogler and Metaxas (1999) proposes PaHMMs which models the movement and shape of user's hands in independent channels and then put them together at the recognition stage. Chen and Koskela (2013) uses multiple Extreme Learning Machines (ELMs) (Huang et al., 2012) as classifiers for simultaneous components. The outputs from the multiple ELMs are then fused and aggregated to provide the final classification results. Chen and Koskela (2013) proposes a novel representation of human gestures and actions based on component concurrent character. They learn the parameters of a statistical distribution that describes the location, shape, and motion flow. Inspired by the sequential organization character of gestures, Wang et al. (2002) uses the segmented subsequences instead of the whole gesture sequence as the basic units that convey the specific semantic expression for the gesture and encode the gesture based on these units. It is successfully applied in large vocabulary sign gestures recognition.

To our best knowledge, there is no work in the literature modeling both the component concurrent character and the sequential organization character in gesture recognition, especially for one-shot learning gesture recognition. It should be noted that these two characters demonstrate the essences of gestures from spatial and temporal aspects, respectively. Therefore, the proposed method that exploits both these characters in a multi-layered framework is desirable to improve gesture recognition.

Test	*Avg. Acc. (%)*	*Identification Strategy*	*Description*
1	75.0	None	Memorizing all the training gestures, and identifying test gesture by recollection
2	90.3	Motion	Drawing lines to record motion direction of each training gesture
3	83.5	Shape	Drawing sketches to describe the hand shape of each training gesture
4	87.6	Location	Drawing sketches to describe the location of each training gesture
5	95.3	Motion & Shape	Strategy 2 and 3
6	100.0	Motion & Location & Shape	Strategy 2, 3 and 4

Table 1: Observations on CGD 2011.

3. Multi-layered Gesture Recognition

The proposed multi-layered classification framework for one-shot learning gesture recognition contains three layers as shown in Figure 1. In the first layer, an improved principle motion is applied to model the motion component. In the second layer, a particle based descriptor is proposed to extract dynamic gesture information and then a weighted dynamic time warping is proposed for the location component classification. In the last layer, we extract unclosed shape contour from the key frame of a gesture sequence. Spatial path warping is further proposed to recognize the shape component. Once the motion component classification at the first layer is accomplished, the original gesture candidates are divided into possible gesture candidates and impossible gesture candidates. The possible gesture candidates are then fed to the second layer which performs the location component classification. Compared with the original gesture candidates, classifying the possible gesture candidates is expected to reduce the computational complexity of the second layer distinctly. The possible gesture candidates are further reduced by the second layer. In the reduced possible gesture candidates, if the first two best matched candidates are difficult to be discriminated, i.e., the absolute difference of their matching scores is lower than a predefined threshold, then the reduced gesture candidates are forwarded to the third layer; otherwise the best matched gesture is output as the final recognition result.

In the remaining of this section, the illuminating cues are first observed in Section 3.1. Inter-gesture segmentation is then introduced in Section 3.2. The motion, location and shape component classifiers in each layer are finally introduced in Section 3.3, Section 3.4 and Section 3.5, respectively.

3.1 Gesture Meaning Expressions and Illuminating Cues

Although from the point of view of gesture linguistics, the basic components and how gestures convey meaning are given (Stokoe, 1960), there is no reference to the importance and complementarity of the components in gesture communication. This section wants to draw some illuminating cues from observations. For this purpose, 10 undergraduate volunteers are invited to take part in the observations.

Five batches of data are randomly selected from the development data of CGD 2011. The pre-defined identification strategies are shown in Table 1. In each test, all the volunteers are asked to follow these identification strategies. For example, in Test 2, they are required to only use the motion cue and draw simple lines to record the motion direction of each gesture in the training set. Then the test gestures are shown to the volunteers to be identified using these drawn lines. The results are briefly summarized in Table 1.

From the observations above, the following illuminating cues can be drawn:

- During gesture recognition, gesture components in the order of importance are motion, location and shape.
- Understanding a gesture requires the observation of all these gesture components. None of these components can convey the complete gesture meanings independently. These gesture components complement each other.

3.2 Inter-gesture Segmentation Based on Movement Quantity

The inter-gesture segmentation is used to segment a multi-gesture sequence into several gesture sequences.[1] To perform the inter-gesture segmentation, we first measure the quantity of movement for each frame in a multi-gesture sequence and then threshold the quantity of movement to get candidate boundaries. Then, a sliding window is adopted to refine the candidate boundaries to produce the final boundaries of the segmented gesture sequences in a multi-gesture sequence.

3.2.1 Quantity of Movement

In a multi-gesture sequence, each frame has the relevant movement with respect to its adjacent frame and the first frame. These movements and their statistical information are useful for inter-gesture segmentation. For a multi-gesture depth sequence I, the Quantity of Movement (QOM) for frame t is defined as a two-dimensional vector

$$QOM(I,t) = [QOM_{Local}(I,t), QOM_{Global}(I,t)] \ ,$$

where $QOM_{Local}(I,t)$ and $QOM_{Global}(I,t)$ measure the relative movement of frame t respective to its adjacent frame and the first frame, respectively. They can be computed as

$$\begin{aligned} QOM_{Local}(I,t) &= \sum_{m,n} \sigma(I_t(m,n), I_{t-1}(m,n)) \ , \\ QOM_{Global}(I,t) &= \sum_{m,n} \sigma(I_t(m,n), I_1(m,n)) \ , \end{aligned}$$

where (m,n) is the pixel location and the indicator function $\sigma(x,y)$ is defined as

$$\sigma(x,y) = \begin{cases} 1 & if|x-y| \geq Threshold_{QOM} \\ 0 & \text{otherwise} \end{cases} \ ,$$

where $Threshold_{QOM}$ is a predefined threshold, which is set to 60 empirically in this paper.

3.2.2 Inter-gesture Segmentation

We assume that there is a home pose between a gesture and another one in a multi-gesture sequence. The inter-gesture segmentation is facilitated by the statistical characteristics of QOM_{Global} of the beginning and ending phases of the gesture sequences in the training

1. In this paper, we use the term "gesture sequence" to mean an image sequence that contains only one complete gesture and "multi-gesture sequence" to mean an image sequence which may contain one or multiple gesture sequences.

data. One advantage of using QOM_{Global} is that it does not need to segment the user from the background.

Firstly the average frame number L of all gestures in the training set is obtained. The mean and standard deviation of QOM_{Global} of the first and last $\lceil L/8 \rceil$ frames of each gesture sequence are computed. After that, a threshold $Threshold_{inter}$ is obtained as the sum of the mean and the doubled standard deviation. For a test multi-gesture sequence T which has t_s frames, the inter-gesture boundary candidate set is defined as

$$B^{ca}_{inter} = \{i | QOM_{Global}(T, i) \leq Threshold_{inter}, i \in \{1, \cdots, t_s\}\} \ .$$

The boundary candidates are further refined through a sliding window of size $\lceil L/2 \rceil$, defined as $\{j+1, j+2, \cdots, j+\lceil L/2 \rceil\}$ where j starts from 0 to $t_s - \lceil L/2 \rceil$. In each sliding window, only the candidate with the minimal QOM_{Global} is retained and other candidates are eliminated from B^{ca}_{inter}. After the sliding window stops, the inter-gesture boundaries are obtained, which are exemplified as the blue dots in Figure 2. The segmented gesture sequences will be used for motion, location, and shape component analysis and classification.

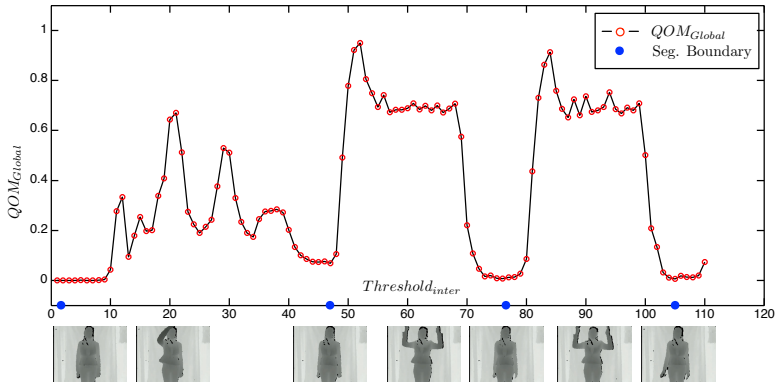


Figure 2: An example of illustrating the inter-gesture segmentation results.

3.3 Motion Component Analysis and Classification

Owing to the relatively high importance of the motion component, it is analyzed and classified in the first layer. The principal motion (Escalante and Guyon, 2012) is improved by using the overlapping block partitioning to reduce the errors of motion pattern mismatchings. Furthermore, our improved principal motion uses both the RGB and depth images. The gesture candidates outputted by the first layer is then fed to the second layer.

3.3.1 Principal Motion

Escalante and Guyon (2012) uses a set of histograms of motion energy information to represent a gesture sequence and implements a reconstruction based gesture recognition method based on principal components analysis (PCA). For a gesture sequence, motion energy images are calculated by subtracting consecutive frames. Thus, the gesture sequence with N frames is associated to $N-1$ motion energy images. Next, a grid of equally spaced

blocks is defined over each motion energy image as shown in Figure 3(c). For each motion energy image, the average motion energy in each of the patches of the grid is computed by averaging values of pixels within each patch. Then a 2D motion map for each motion energy image is obtained and each element of the map accounts for the average motion energy of the block centered on the corresponding 2D location. The 2D map is then vectorized into an N_b-dimensional vector. Hence, an N frame gesture sequence is associated to a matrix Y of dimensions $(N-1) \times N_b$. All gestures in the reference set with size V can be represented with matrices Y_v, $v \in \{1, \cdots, V\}$ and PCA is applied to each Y_v. Then the eigenvectors corresponding to the top c eigenvalues form a set $\mathcal{W}_v$, $v = \{1, \cdots, V\}$.

In the recognition stage, each test gesture is processed as like training gestures and represented by a matrix S. Then, S is projected back to each of the V spaces induced by $\mathcal{W}_v$, $v \in \{1, \cdots, V\}$. The V reconstructions of S are denoted by $R_1, \cdots, R_\mathcal{V}$. The reconstruction error of each R_v is computed by

$$\varepsilon(v) = \frac{1}{n} \sum_{i=1}^{n} \sqrt{\sum_{j=1}^{m} (R_v(i,j) - S(i,j))^2} \ ,$$

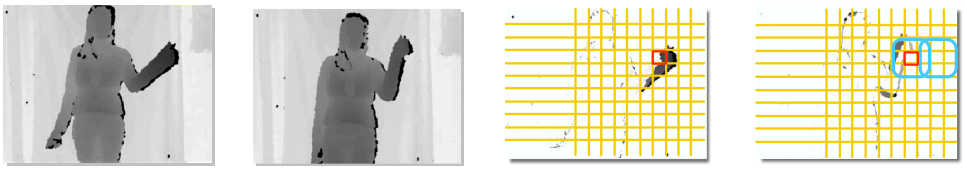
where n and m are the number of rows and columns of S. Finally, the test gesture is recognized as the gesture with label obtained by $\arg\min_v \varepsilon(v)$.

3.3.2 Improved Principle Motion

Gestures with large movements are usually performed with significant deformation as shown in Figure 3. In Escalante and Guyon (2012), motion information is represented by a histogram whose bins are related to spatial positions. Each bin is analyzed independently and the space interdependency among the neighboring bins is not further considered. The interdependency can be explored to improve the robustness of representing the gesture motion component, especially for the gestures with larger movement. To this end, an overlapping neighborhood partition is proposed. For example, if the size of bins is 20×20, the overlapping neighborhood contains 3×3 equally spaced neighboring bins in a 60×60 square region. The averaged motion energy in the square region is taken as the current bin's value as shown in Figure 3.

The improved principle motion is applied to both the RGB and depth data. The RGB images are transformed into gray images before computing their motion energy images. For each reference gesture, the final V reconstruction errors are obtained by multiplying the reconstruction errors of the depth data and the gray data. These V reconstruction errors are further clustered by K-means to get two centers. The gesture labels associated to those reconstruction errors belonging to the center with smaller value are treated as the possible gesture candidates. The remaining gesture labels are treated as the impossible gesture candidates. Then the possible candidates are fed to the second layer.

We compare the performance of our improved principal motion model with the original principal motion model (Escalante and Guyon, 2012) on the first 20 development batches of CGD 2011. Using the provided code (Guyon et al., 2014; Escalante and Guyon, 2012) as baseline, the average Levenshtein distances (Levenshtein, 1966) are 44.92% and 38.66% for the principal motion and the improved principal motion, respectively.



(a) (b) (c) (d)

Figure 3: An example of a gesture with large movements. (a) and (b): two frames from a gesture. (c): the motion energy image of (a). The grid of equally spaced bins adopted by the Principle Motion (Escalante and Guyon, 2012). (d): the motion energy image of (b). The overlapped grid used by our method where the overlapping neighborhood includes all 3×3 equally spaced neighbor bins.

3.4 Location Component Analysis and Classification

Gesture location component refers to the positions of the arms and hands relative to the body. In the second layer, the sequential organization character of gestures is utilized in the gesture sequence alignment. According to the movement-hold model, each gesture sequence is segmented into semantic units, which convey the specific semantic meanings of the gesture. Accordingly, when aligning a reference gesture and a test gesture, the semantic units are aligned first, then the frames in each semantic unit are registered. A particle-based representation for the gesture location component is proposed to describe the location component of the aligned frames and a Weighted Dynamic Time Warping (WDTW) is proposed for the location component classification.

3.4.1 Intra-gesture Segmentation and Alignment

To measure the distance between location components of a reference gesture sequence $R = \{R_1, R_2 \cdots, R_{L_R}\}$ and a test gesture sequence $T = \{T_1, T_2 \cdots, T_{L_T}\}$, an alignment $\Gamma = \{(i_k, j_k) | k = 1, \cdots, K, i_k \in \{1, \cdots, L_R\}, j_k \in \{1, \cdots, L_T\}\}$ can be determined by the best path in the Dynamic Time Warping (DTW) grid and K is the path length. Then the dissimilarity between two gesture sequences can be obtained as the sum of the distances between the aligned frames.

The above alignment does not consider the sequential organization character of gestures. The movement-hold model proposed by Liddell and Johnson (1989) reveals sequential organization of gestures, which should be explored in the analysis and classification of gesture location component. $QOM_{Local}(I, t)$, described in Section 3.2.1, measures the movement between two consecutive frames. A large $QOM_{Local}(I, t)$ indicates that the t-th frame is in a movement phase, while a small $QOM_{Local}(I, t)$ indicates that the frame is in a hold phase. Among all the frames in a hold phase, the one with the minimal $QOM_{Local}(I, t)$ is the most representative frame and is marked as an anchor frame. Considering the sequential organization character of gestures, the following requirement should be satisfied to compute Γ: each anchor frame in a test sequence must be aligned with one anchor frame in the reference sequence.

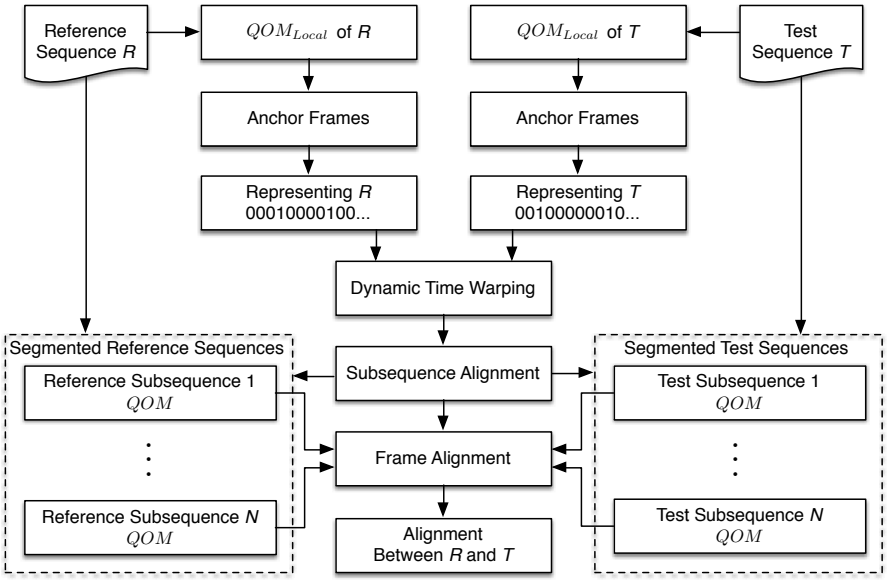


Figure 4: Intra-gesture segmentation and the alignment between test and reference sequences.

As shown in Figure 4, the alignment between the test and reference sequences has two stages. In the first stage, DTW is applied to align the reference and test sequences. Each anchor frame is represented by "1" and the remaining frames are represented by "0". Then the associated best path $\widehat{\Gamma} = \{(\widehat{i_k}, \widehat{j_k}) | k = 1, \cdots, \widehat{K}\}$ in the DTW grid can be obtained. For each $(\widehat{i_k}, \widehat{j_k})$, if both $\widehat{i_k}$ and $\widehat{j_k}$ are anchor frames, then $\widehat{i_k}$ and $\widehat{j_k}$ are the boundaries of the semantic units. According to the boundaries, the alignment between semantic units of the reference and test sequences is obtained. In the second stage, as shown in Figure 4, each frame in a semantic unit is represented by $[QOM_{Local}, QOM_{Global}]$ and DTW is applied to align the semantic unit pairs separately. Then the final alignment Γ is obtained by concatenating the alignments of the semantic unit pairs.

3.4.2 Location Component Segmentation and its Particle Representation

After the frames of the test and reference sequences are aligned, the next problem is how to represent the location information in a frame. Dynamic regions in each frame contain the most meaningful location information, which are illustrated in Figure 5(i).

A simple thresholding-based foreground-background segmentation method is used to segment the user in a frame. The output of the segmentation is a mask frame that indicates which pixels are occupied by the user as shown in Figure 5(b). The mask frame is then denoised by a median filter to get a denoised frame as shown in Figure 5(c). The denoised frame is first binarized and then dilated with a flat disk-shaped structuring element with radius 10 as shown in Figure 5(d). The swing frame as shown in Figure 5(h) is obtained by subtracting the binarized denoised frame from the dilated frame. The swing region (those

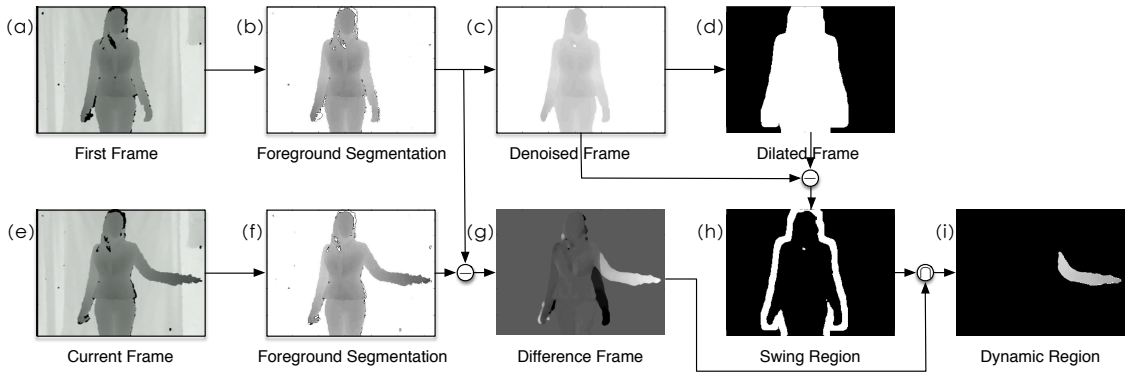


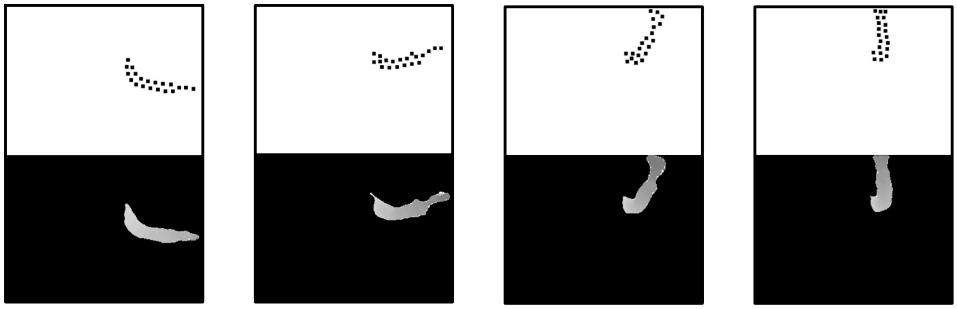
Figure 5: Dynamic region segmentation.

white pixels in the swing frame) covers the slight swing of user's trunk and can be used to eliminate the influence of body swing. From frame t, define set Ξ as

$$\{(m,n)|F_1(m,n) - F_t(m,n) \geq Threshold_{QOM}\} \ ,$$

where F_1 and F_t are the user masks of the first frame and frame t, respectively. $Threshold_{QOM}$ is the same as in Section 3.2.1. For each connected region in Ξ, only if the number of pixels in this region exceeds N_p and the proportion overlapped with swing region is less than r, it is regarded as a dynamic region. Here $N_p = 500$ is a threshold used to remove the meaningless connected regions in the difference frame as shown in Figure 5(g). If a connected region has less than N_p pixels, we think this region should not be a good dynamic region for extracting location features, e.g., the small bright region on the right hand of the user in Figure 5(g). This parameter can be set intuitively. The parameter $r = 50\%$ is also a threshold used to complement with N_p to remove the meaningless connected regions in the difference frame. After using N_p to remove some connected regions, there may be a retained connected region which has more than N_p pixels but it may still not be a meaningful dynamic region for extracting position features if the connected region is caused by the body swing. Obviously we can exploit the swing region to remove such a region. To do this, we first compute the overlap rate between this region and the swing region. If the overlap rate is larger than r, it is reasonable to think this region is mainly produced by the body swing. Therefore, it should be further removed. As like N_p, this parameter is also very intuitive to set and is not very sensitive to the performance.

To represent the dynamic region of frame t, a particle-based description is proposed to reduce the matching complexity. The dynamic region of frame t can be represented by a 3D distribution: $P_t(x,y,z)$ where x and y are coordinates of a pixel and $z = I_t(x,y)$ is the depth value of the pixel. In the form of non-parametric representation, $P_t(x,y,z)$ can be represented by a set of $\widehat{N}$ particles, $P_{Location}(I_t) = \{(x_n, y_n, z_n)|_{n=1}^{\widehat{N}}\}$. We use K-means to cluster all pixels inside the dynamic region into $\widehat{N}$ clusters. Note that for a pixel, both its spatial coordinates and depth value are used. Then the centers of clusters are used as the representative particles. In this paper, 20 representative particles are used for each frame, as shown in Figure 6.



(a) (b) (c) (d)

Figure 6: Four examples of particle representation of the location component (the black dots are the particles projected onto X-Y plane).

3.4.3 Location component Classification

Assume the location component of two aligned frames can be represented as two particle sets, $P = \{P_1, P_2 \cdots P_{\widehat{N}}\}$ and $Q = \{Q_1, Q_2 \cdots Q_{\widehat{N}}\}$. The matching cost between particle P_i and Q_j, denoted by $C(P_i, Q_j)$, is computed as their Euclidean distance. The distance of the location component between these two aligned gesture frames is defined by the minimal distance between P and Q. Computing the minimal distance between two particle sets is indeed to find an assignment Π to minimize the cost summation of all particle pairs

$$\Pi = \arg\min_{\Pi} \sum_{i=1}^{\widehat{N}} C(P_i, Q_{\Pi(i)}) \ . \tag{1}$$

This is a special case of the weighted bipartite graph matching and can be solved by the Edmonds method (Edmonds, 1965). Edmonds method which finds an optimal assignment for a given cost matrix is an improved Hungarian method (Kuhn, 1955) with time complexity $O(n^3)$ where n is the number of particles. Finally, the distance of the location component between two aligned gesture frames is obtained

$$dis(P, Q) = \sum_{i=1}^{\widehat{N}} C(P_i, Q_{\Pi(i)}) \ .$$

The distance between the reference sequence R and the test sequence T can be computed as the sum of all distance between the location components of the aligned frames in Γ

$$DIS_{Location}(R, T|\Gamma) = \sum_{k=1}^{K} dis(P_{Location}(R_{i_k}), P_{Location}(T_{j_k})) \ . \tag{2}$$

This measurement implicitly gives all the frames the same weights. However, in many cases gestures are distinguished by only a few frames. Therefore, rather than directly computing

Equation 2, we propose the Weighted DTW (WDTW) to compute the distance of location component between R and T as

$$WDIS_{Location}(R,T|\Gamma) = \sum_{k=1}^{K} W_{i_k}^R \times dis(P_{Location}(R_{i_k}), P_{Location}(T_{j_k})) \ ,$$

where $W^R = \{W_{i_k}^R | i_k \in \{1, \cdots, L_R\}\}$ is the weight vector. Different from the method of evaluating the phase difference between the test and reference sequences (Jeong et al., 2011) and the method of assigning different weights to features (Reyes et al., 2011), we assign different weights to the frames of the reference gesture sequence. For each reference gesture sequence, firstly we use the regular DTW to calculate and record the alignment Γ between the current reference gesture sequence and all the other reference gesture sequences. Secondly for each frame in the current reference gesture sequence, we accumulate its corresponding distances with the matched frames in the best path in the DTW. Then, the current frame is weighted by the average distance between itself and all the corresponding frames in the best path. The detailed procedure of computing the weight vector are summarized in Algorithm 1.

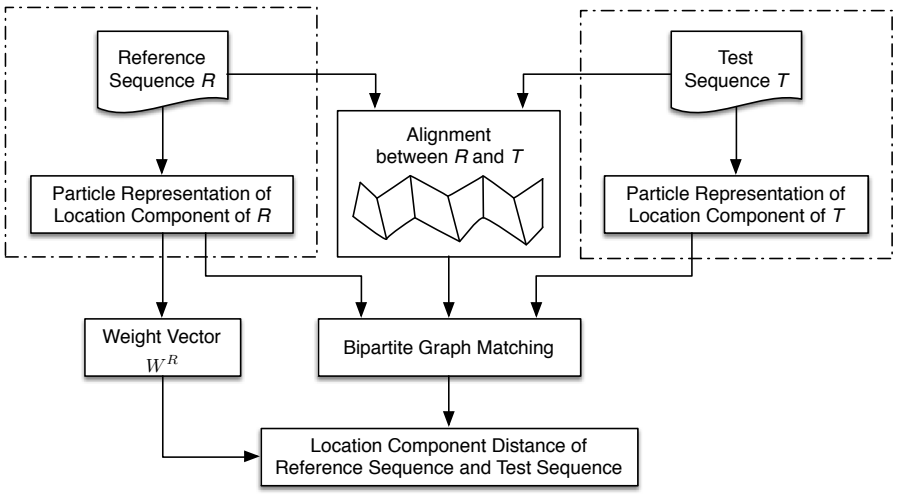


Figure 7: Weighted Dynamic Time Warping framework.

In the second layer, we first use K-means to cluster the input possible gesture candidates into two cluster centers according to the matching scores between the test gesture sequence and the possible gesture candidates. The candidates in the cluster with smaller matching score are discarded. In the remaining candidates, if the first two best matched candidates are difficult to be distinguished, i.e., the absolute difference of their normalized location component distances is lower than a predefined threshold ϵ, then these candidates are forwarded to the third layer; otherwise the best matched candidate is output as the final recognition result. Two factors influence the choice of the parameter ϵ. The first one is the number of the gesture candidates and the other one is the type of gestures. When the number of the gesture candidates is large or most of the gesture candidates are the shape

Algorithm 1 Computing weight vector W^R for a reference R

Input: all the O reference gesture depth sequences: $I^1, I^2, \cdots, I^O$
Output: weight vector for R, $W^R = \{W_m^R | m \in \{1, \cdots, L_R\}\}$

for each $m \in [1, L_R]$ **do**
  $W_m^R = 0$
  $N_m^R = 0$
end for
for each $n \in [1, O]$ **do**
  Compute the alignment $\Gamma = \{(i_k, j_k)\}$ between R and I^n
  for each $m \in [1, L_R]$ **do**
    $W_m^R = W_m^R + \sum_{(i_k=m, j_k)\in\Gamma} dis(P_{Location}(R_{i_k}), P_{Location}(I_{j_k}^n))$
    $N_m^R = N_m^R + \sum_{(i_k, j_k)\in\Gamma} \delta(i_k = m)$
    if $n = O$ **then**
      $W_m^R = W_m^R / N_m^R$
    end if
  end for
end for

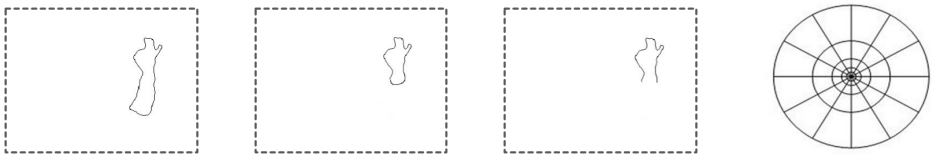
dominant gestures, a high threshold is preferred. In our experiments, we empirically set its value with 0.05 by observing the matching scores between the test sample and each gesture candidates.

3.5 Shape Component Analysis and Classification

The shape in a hold phase is more discriminative than the one in a movement phase. The key frame in a gesture sequence is defined as the frame which has the minimization QOM_{Local}. Shape component classifier classifies the shape features extracted from the key frame of a gesture sequence using the proposed Spatial Path Warping (SPW), which first extracts unclosed shape context (USC) features and then calculates the distance between the USCs of the key frames in the reference and the test gesture sequences. The test gesture sequence is classified as the gesture whose reference sequence has the smallest distance with the test gesture sequence.

3.5.1 Unclosed Shape Segmentation

The dynamic regions of a frame have been obtained in Section 3.4.2. In a key frame, the largest dynamic region D is used for shape segmentation. Although shapes are complex and do not have robust texture and structured appearance, in most cases shapes can be distinguished by their contours. The contour points of D are extracted by the Canny algorithm. The obtained contour point set is denoted by C_1 as shown in Figure 8(a). K-means is adopted to cluster the points in D into two clusters based on the image coordinates and depth of each point. If a user faces to the camera, the cluster with smaller average depth contains most of information for identifying the shape component. Canny algorithm is used again to extract contour points of the cluster with smaller average depth. The obtained closed contour point set is denoted by C_2 as shown in Figure 8(b). Furthermore, an unclosed contour point set can be obtained by $C_3 = C_2 \bigcap C_1$ as shown in Figure 8(c), which will be used to reduce the computational complexity of matching shapes.



(a) (b) (c) (d)

Figure 8: Unclosed shape segmentation and context representation. (a) is an example of point set C_1, (b) is an example of point set C_2 and (c) is an example of obtained point set C_3; (d) is the log-polar space used to decide the ranges of K bins.

3.5.2 Shape Representation and Classification

The contour of a shape consists of a 2-D point set $\mathbb{P} = \{p_1, p_2, \cdots, p_N\}$. Their relative positions are important for the shape recognition. From the statistical point of view, Belongie et al. (2002) develops a strong shape contour descriptor, namely Shape Context (SC). For each point p_i in the contour, a histogram h_{p_i} is obtained as the shape context of the point whose k-th bin is calculated by

$$h_{p_i}(k) = \sharp\{(p_j - p_i) \in bin(k) | p_j \in \mathbb{P}, i \neq j, k \in \{1, \cdots, K\}\} \ ,$$

where $bin(k)$ defines the quantification range of the k-th bin. The log-polar space for bins is illustrated in Figure 8(d).

Assume $\mathbb{P}$ and $\mathbb{Q}$ are the point sets for the shape contours of two key frames, the matching cost $\Phi(p_i, q_j)$ between two points $p_i \in \mathbb{P}$ and $q_j \in \mathbb{Q}$ is defined as

$$\Phi(p_i, q_j) = \frac{1}{2} \sum_{k=1}^{K} \frac{[h_{p_i}(k) - h_{q_j}(k)]^2}{h_{p_i}(k) + h_{q_j}(k)} \ .$$

Given the set of matching costs between all pairs of points $p_i \in \mathbb{P}$ and $q_j \in \mathbb{Q}$, computing the minimal distance between $\mathbb{P}$ and $\mathbb{Q}$ is to find a permutation Ψ to minimize the following sum

$$\Psi = \arg\min_{\Psi} \sum_{i} \Phi(p_i, q_{\Psi(i)}) \ ,$$

which can also be solved by the Edmonds algorithm as like solving Equation 1.

An unclosed contour contains valuable spatial information. Thus, a Spatial Path Warping algorithm (SPW) is proposed to compute the minimal distance between two unclosed contours. Compared with the Edmonds algorithm, the time complexity of the proposed SPW is reduced from $O(n^3)$ to $O(n^2)$ where n is the size of the point set of an unclosed shape contour. As shown in Figure 8(c), the points on an unclosed contour can be represented as a clockwise contour point sequence. SPW is used to obtain the optimal match between two given unclosed contour point sequences. For two unclosed contour point sequences $\{p'_1, \cdots, p'_n\}$, $\{q'_1, \cdots, q'_m\}$, a dynamic window is set to constrain the points that one point can match, which makes the matching more robust to local shape variation. We

set the window size w with $\max(L_s, abs(n-m))$. In most cases, the window size is the absolute difference between the lengths of the two point sequences. In extreme cases, if two sequences have very close lengths, i.e., their absolute difference is less then L_s, we set the the window size with L_s. The details of proposed SPW are summarized in Algorithm 2.

Algorithm 2 Computing distance between two unclosed contour point sequences

Input: two unclosed contour point sequences $\{p'_1, \cdots, p'_n\}$, $\{q'_1, \cdots, q'_m\}$
Output: distance between these two point sequences $SPW[n, m]$.

Set $w = \max(L_s, abs(n-m))$
for each $i \in [0, n]$ **do**
  for each $j \in [0, m]$ **do**
    $SPW[i, j] = \infty$
  end for
end for
$SPW[0, 0] = 0$
for each $i \in [1, n]$ **do**
  for each $j \in [\max(1, i-w), \min(m, i+w)]$ **do**
    $SPW[i, j] = \Phi(p'_i, q'_j) + \min(SPW[i-1, j], SPW[i, j-1], SPW[i-1, j-1])$
  end for
end for

4. Experiments

In this section, extensive experiment results are presented to evaluate the proposed multi-layered gesture recognition method. All the experiments are performed in Matlab 7.12.0 on a Dell PC with Duo CPU E8400. The ChaLearn Gesture Dataset (CGD 2011) (Guyon et al., 2014) is used in all experiments, which is designed for one-shot learning. The CGD 2011 consists of 50,000 gestures (grouped in 500 batches, each batch including 47 sequences and each sequence containing 1 to 5 gestures drawn from one of 30 small gesture vocabularies of 8 to 15 gestures), with frame size 240×320, 10 frames/second, recorded by 20 different users.

The parameters used in the proposed method are listed in Table 2. Noted that the parameters c and N_b are set with the default values used in the sample code of the principal model.[2] The threshold for foreground and background segmentation is adaptively set to the maximal depth minus 100 for each batch data. For example, the maximal depth of the devel01 batch is 1964. Then the threshold for this batch is 1864. The number 100 is in fact a small bias from the maximal depth, which is empirically set in our experiments. We observed that slightly changing this number does not significantly affect the segmentation. Considering the tradeoff between the time complexity and recognition accuracy, in our experiments, we empirically set $\widehat{N}$ to 20, which achieves the desired recognition performance.

In our experiments, Levenshtein distance is used to evaluate the gesture recognition performance, which is also used in the CHALERAN gesture challenge. It is the minimum number of edit operations (substitution, insertion, or deletion) that have to be performed from one sequence to another (or vice versa). It is also known as "edit distance".

2. The code is available at `http://gesture.chalearn.org/data/sample-code`

Parameter and Description	*Applied to*	*Value*	*From Prior or Not*	*Sensitive to Performance*	*Training Data Used or Not*
N_p: Minimal number of pixels in a connected region	**D**	500	Y	N	Y
r: Maximal overlap rate between a connected region and the swing region	D	50%	N	N	N
ϵ: Threshold for the difference between the first two largest matches	**D**, **E**	0.05	Y	N	Y
L_s: Minimal length of the sliding window	**E**	5	N	N	N
$Threshold_{QOM}$	**A**, **D**, **E**	60	Y	Y	N
$Threshold_{inter}$	**A**	adaptive	N	Y	Y
c: number of eigenvalues for each gesture	**C**	10	Y	N	N
N_b: number of bins for each motion energy image	**C**	192	Y	N	N
$\hat{N}$: number of particles	**D**	20	Y	N	N
Threshold for foreground and background segmentation	**D**, **E**	Max depth - 100	Y	N	Y
A: Inter-gesture segmentation; **B**: intra-gesture segmentation; **C**: Motion component analysis and classification **D**: Location component analysis and classification; **E**: Shape component analysis and classification; **Training data**: CGD 2011					

Table 2: The parameters used in the proposed multi-layered gesture recognition and their descriptions.

4.1 Performance of Our Method with Different Layers

We evaluate the performance of the proposed method with different layers on the development (devel01 $\sim$ devel480) batches of CGD 2011 and Table 3 reports the results. If only the first layer is used for classification, the average Levenhstein distance is 37.53% with running time 0.54 seconds per gesture. If only the second layer is used for recognition, the average Levenhstein distance is 29.32% with running time 6.03 seconds per gesture. If only the third layer is used, the average Levenhstein distance is 39.12% with the running time 6.64 seconds per gesture. If the first two layers are used, the average Levenhstein distance is 24.36% with running time 2.79 seconds per gesture. If all three layers are used, the average normalized Levenhstein distance is 19.45% with running time 3.75 seconds per gesture.

methods	*First layer for recognition*	*Second layer for recognition*	*Third layer for recognition*	*First two layers for recognition*	*Three layers for recognition*
TeLev (%)	37.53	29.32	39.12	24.36	19.45
Recognition time per gesture (s)	0.54	6.03	6.64	2.79	3.75

Table 3: Performance of using the first layer, the second layer, the third layer, first two layers and three layers on ChaLearn gesture data set (devel01 $\sim$ devel480).

From these comparison results, we can see that the proposed method achieves high recognition accuracy while having low computational complexity. The first layer can identify the gesture candidates at the speed of 80 fps (frames per second). The second layer has relatively high computational complexity. If we only use the second layer for classification, the average computing time is roughly 11 times of the first layer. Despite with relatively high computational cost, the second layer has stronger classification ability. Compared with using only the second layer, the computational complexity of using the first two layers in the proposed method is distinctly reduced and can achieve 16 fps. The reason is that although the second layer is relatively complex, the gesture candidates forwarded to it are significantly reduced by the first layer. When all three layers are used, the proposed method still achieve about 12 fps, which is faster than the video recording speed (10 fps) of CGD 2011.

4.2 Comparison with Recent Representative Methods

We compare the proposed method with other recent representative methods on the first 20 development data batches. Table 4 reports the performance of the proposed method on each batch and also the average performance on all 20 batches. The average performance of the proposed method and the compared methods are shown in Table 5.

Batch	Second layer for recognition		First two layers for recognition		Three layers for recognition	
	TeLev (%)	***Recognize time per gesture (s)***	***TeLev (%)***	***Recognize time per gesture (s)***	***TeLev (%)***	***Recognize time per gesture (s)***
1	7.24	6.78	0.11	3.40	1.11	3.59
2	41.21	11.38	44.21	7.10	34.35	10.00
3	62.98	8.86	69.20	2.99	39.95	5.61
4	4.51	5.98	3.93	2.10	6.93	2.30
5	11.68	10.96	2.62	3.05	4.77	3.31
6	44.64	5.59	39.94	2.69	23.51	3.42
7	12.44	3.59	8.51	1.70	8.51	1.79
8	5.56	4.94	0.00	2.14	5.71	2.94
9	10.56	5.10	6.44	2.50	6.44	3.01
10	44.21	5.88	29.13	3.24	16.52	3.95
11	42.75	6.46	36.36	3.98	28.93	6.31
12	8.56	5.16	1.06	2.00	7.06	2.34
13	16.24	3.68	12.93	1.20	12.93	1.99
14	44.69	2.50	40.13	0.90	27.98	2.35
15	15.78	4.61	4.21	1.09	6.21	2.19
16	36.54	8.35	36.27	4.21	23.41	6.94
17	36.25	9.10	29.55	5.10	26.32	5.39
18	62.4	1.99	69.21	0.81	53.55	1.60
19	54.31	5.07	51.32	2.84	47.61	3.02
20	17.74	2.58	10.61	1.40	10.61	2.01
Average	**29.02**	**5.93**	**24.79**	**2.73**	**19.62**	**3.69**

Table 4: Recognition performance of using the second layer, first two layers and three layers on first 20 development batches of CGD 2011 (TeLev is the average Levenshtein distance).

Methods	***Extend-MHI Wu et al. (2012a)***	***Manifold LSR Lui (2012a)***	***Sparse Coding Fanello et al. (2013)***	***Temporal Bayesian Malgireddy et al. (2012)***	***Motion History Mahbub et al. (2013)***	***CSMMI+3D EMoSIFT Wan et al. (2014a)***	***Proposed***
TeLev (%)	26.00	28.73	25.11	24.09	31.25	18.76	19.62
TeLen	#	6.24	5.02	#	18.01	#	5.91

Table 5: Performance comparison on the 20 development data batches (TeLen is the average error made on the number of gestures).

For the comparison on each batch, the proposed method is compared with a manifold and nonlinear regression based method (Manifold LSR) (Lui, 2012b), an extended motion-history-image and correlation coefficient based method (Extended-MHI) (Wu et al., 2012a), and a motion silhouettes based method (Motion History) (Mahbub et al., 2013). The comparison results are shown in Figure 9.

In batches 13, 14, 17, 18, 19, the proposed method does not achieve the best performance. However, the proposed method achieves the best performance in the remaining 15 batches.

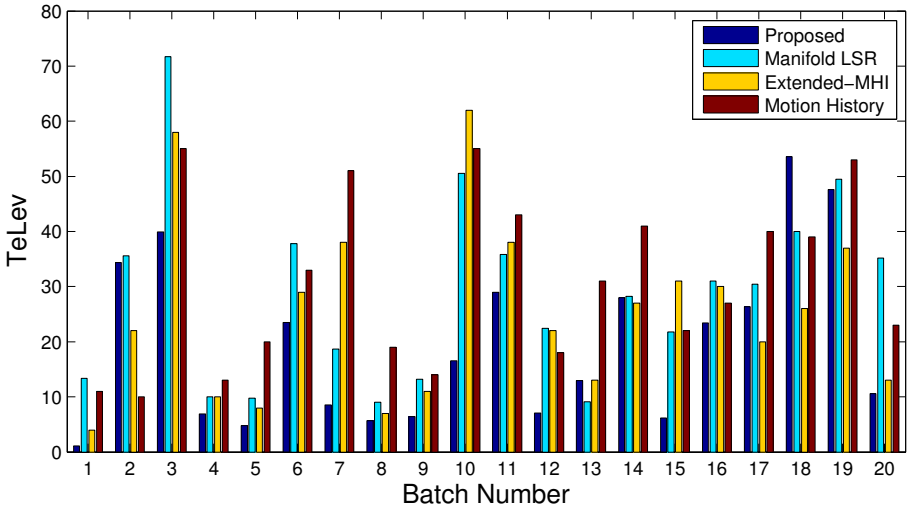


Figure 9: Performance comparison on the 20 development batches in CGD 2011.

In batches 3, 10 and 11, most of gestures consist of static shapes, which can be efficiently identified by the shape classifier in the third layer. Batches 1, 4, 7 and 8 consist of motion dominated gestures, which can be classified by the motion and location component classifiers in the first and second layers. In batches 18 and 19, the proposed method has relatively poor performance. As in batch 18, most of gestures have small motion, similar locations, and non-stationary hand shapes. These gestures may be difficult to be identified by the proposed method. In batch 19, the gestures have similar locations and hands coalescence, which is difficult to be identified by the second layer and the third layer classifiers in our method. Overall, the proposed method significantly outperforms other recent competitive methods.

The proposed method is further compared with DTW, continuous HMM (CHMM), semi-continuous HMM (SCHMM) and SOFM/HMM (Fang et al., 2004) on the development (devel01 $\sim$ devel480) batches of CGD 2011. All compared methods use one of three feature descriptors including dynamic region grid representation (DP), dynamic region particle representation (DG) and Dynamic Aligned Shape Descriptor (DS) (Fornés et al., 2010).

- **Dynamic region grid representation**. For the dynamic region of the current frame obtained in Section 3.4.2, a grid of equally spaced cells is defined and the default size of grid is 12×16. For each cell, the average value of depth in the square region is taken as the value of current bin. So a 12×16 matrix is generated, which is vectorized into the feature vector of the current frame.
- **Dynamic region particle representation**. The particles for the current frame obtained in Section 3.4.2 cannot directly be used as an input feature vector and they have to be reorganized. The 20 particles $\{(x_n, y_n, z_n)|_{n=1}^{20}\}$ are sorted according to $\|(x_n, y_n)\|^2$ and then the sorted particles are concatenated in order to get a 60-dimensional feature vector to represent the current frame.

- **Dynamic region D-Shape descriptor** (Fornés et al., 2010). Firstly, the location of some concentric circles is defined, and for each one, the locations of the equidistant voting points are computed. Secondly, these voting points will receive votes from the pixels of the shape of the dynamic region, depending on their distance to each voting point. By locating isotropic equidistant points, the inner and external part of the shape could be described using the same number of voting points. In our experiment, we used 11 circles for the D-Shape descriptor. Once we have the voting points, the descriptor vector is computed.

Here, each type of HMM is a 3-state left-to-right model allowing possible skips. For CHMM and SCHMM, the covariance matrix is a diagonal matrix with all diagonal elements being 0.2. The comparison results are reported in Table 6.

Method	***Number of Mixtures for each state***	***TeLev (%)***			***Recognition time per gesture (s)***		
		DP	***DG***	***DS***	***DP***	***DG***	***DS***
DTW	#	38.23	41.19	33.16	2.67	2.51	2.60
CHMM	5	31.41	33.29	31.13	6.91	6.83	6.89
SCHMM	30	31.01	32.92	29.35	6.82	6.75	6.79
SOFM/HMM	5	28.27	30.31	27.20	6.77	6.71	6.74
DP: dynamic region particle representation; **DG**: dynamic region grid representation **DS**: dynamic region D-Shape descriptor							

Table 6: Performance of different sequence matching methods on 480 development batches of CGD 2011.

Compared with these methods, the proposed method achieves the best performance. Noted that in all compared methods, SOFM/HMM classifier with the DS descriptor achieves the second best performance. As explained in Section 1, sequentially modeling motion, position and shape components is very important for improving the performance of gesture recognition. Except the proposed method, other compared methods do not utilize these components. On the other hand, statistical models like CHMM, SCHMM and SOFM/HMM need more training samples to estimate model parameters, which also affect their performance in the one-shot learning gesture recognition.

5. Conclusion

The challenges of gesture recognition lie in the understanding of the unique characters and cues of gestures. This paper proposed a novel multi-layered gesture recognition with Kinect, which is linguistically and perceptually inspired by the phonological model and the movement-hold model. Together with the illuminating cues drawn from observations, the component concurrent character and the sequential organization character of gestures are all utilized in the proposed method. In the first layer, an improved principle motion is applied to model the gesture motion component. In the second layer, a particle based descriptor is proposed to extract dynamic gesture information and then a weighted dynamic

time warping is proposed to classify the location component. In the last layer, the spatial path warping is further proposed to classify the shape component represented by unclosed shape context, which is improved from the original shape context but needs lower matching time. The proposed method can obtain relatively high performance for one-shot learning gesture recognition. Our work indicates that the performance of gesture recognition can be significantly improved by exploring and utilizing the unique characters of gestures, which will inspire other researcher in this field to develop learning methods for gesture recognition along this direction.

Acknowledgments

We would like to acknowledge the editors and reviewers, whose valuable comments greatly improved the manuscript. Specially, we would also like to thank Escalante and Guyon who kindly provided us the principal motion source code and Microsoft Asian who kindly provided two sets of Kinect devices. This work was supported in part by the Major State Basic Research Development Program of China (973 Program 2015CB351804) and the National Natural Science Foundation of China under Grant No. 61272386, 61100096 and 61300111.

References

Tushar Agrawal and Subhasis Chaudhuri. Gesture recognition using motion histogram. In *Proceedings of the Indian National Conference of Communications*, pages 438–442, 2003.

Omar Al-Jarrah and Alaa Halawani. Recognition of gestures in Arabic sign language using neuro-fuzzy systems. *Artificial Intelligence*, 133(1):117–138, 2001.

George Awad, Junwei Han, and Alistair Sutherland. A unified system for segmentation and tracking of face and hands in sign language recognition. In *Proceedings of the 18th International Conference on Pattern Recognition*, volume 1, pages 239–242, 2006.

Malek Baklouti, Eric Monacelli, Vincent Guitteny, and Serge Couvet. Intelligent assistive exoskeleton with vision based interface. In *Proceedings of the 5th International Conference On Smart Homes and Health Telematics*, pages 123–135, 2008.

Britta Bauer and Karl-Friedrich Kraiss. Video-based sign recognition using self-organizing subunits. In *Proceedings of the 16th International Conference on Pattern Recognition*, volume 2, pages 434–437, 2002.

Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002.

Qin Cai, David Gallup, Cha Zhang, and Zhengyou Zhang. 3D deformable face tracking with a commodity depth camera. In *Proceedings of the 11th European Conference on Computer Vision*, pages 229–242, 2010.

Xi Chen and Markus Koskela. Online RGB-D gesture recognition with extreme learning machines. In *Proceedings of the 15th ACM International Conference on Multimodal Interaction*, pages 467–474, 2013.

Helen Cooper, Brian Holt, and Richard Bowden. Sign language recognition. In *Visual Analysis of Humans*, pages 539–562, 2011.

Helen Cooper, Eng-Jon Ong, Nicolas Pugeault, and Richard Bowden. Sign language recognition using sub-units. *Journal of Machine Learning Research*, 13:2205–2231, 2012.

Andrea Corradini. Real-time gesture recognition by means of hybrid recognizers. In *Proceedings of International Gesture Workshop on Gesture and Sign Languages in Human-Computer Interaction*, pages 34–47, 2002.

Ross Cutler and Matthew Turk. View-based interpretation of real-time optical flow for gesture recognition. In *Proceedings of the 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition*, pages 416–416, 1998.

Nasser Dardas. *Real-time Hand Gesture Detection and Recognition for Human Computer Interaction*. PhD thesis, University of Ottawa, 2012.

Paul Doliotis, Alexandra Stefan, Chris Mcmurrough, David Eckhard, and Vassilis Athitsos. Comparing gesture recognition accuracy using color and depth information. In *Proceedings of the 4th International Conference on PErvasive Technologies Related to Assistive Environments*, page 20, 2011.

Jack Edmonds. Maximum matching and a polyhedron with 0, 1-vertices. *Journal of Research of the National Bureau of Standards B*, 69:125–130, 1965.

H Ershaed, I Al-Alali, N Khasawneh, and M Fraiwan. An Arabic sign language computer interface using the Xbox Kinect. In *Proceedings of the Annual Undergraduate Research Conference on Applied Computing*, volume 1, 2011.

Hugo Escalante and Isabelle Guyon. Principal motion. `http://www.causality.inf.ethz.ch/Gesture/principal_motion.pdf`, 2012.

Hugo Jair Escalante, Isabelle Guyon, Vassilis Athitsos, Pat Jangyodsuk, and Jun Wan. Principal motion components for gesture recognition using a single-example. *arXiv preprint arXiv:1310.4822*, 2013.

Sean Ryan Fanello, Ilaria Gori, Giorgio Metta, and Francesca Odone. One-shot learning for real-time action recognition. In *Pattern Recognition and Image Analysis*, pages 31–40, 2013.

Gaolin Fang, Wen Gao, and Debin Zhao. Large vocabulary sign language recognition based on fuzzy decision trees. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 34(3):305–314, 2004.

Alicia Fornés, Sergio Escalera, Josep Lladós, and Ernest Valveny. Symbol classification using dynamic aligned shape descriptor. In *Proceedings of the 20th International Conference on Pattern Recognition*, pages 1957–1960, 2010.

Isabelle Guyon, Vassilis Athitsos, Pat Jangyodsuk, Hugo Jair Escalante, and Ben Hamner. Results and analysis of the Chalearn gesture challenge 2012. In *Proceedings of International Workshop on Advances in Depth Image Analysis and Applications*, pages 186–204, 2013.

Isabelle Guyon, Vassilis Athitsos, Pat Jangyodsuk, and Hugo Jair Escalante. The chalearn gesture dataset (CGD 2011). *Machine Vision and Applications*, 2014. DOI: 10.1007/s00138-014-0596-3.

Chung-Lin Huang and Wen-Yi Huang. Sign language recognition using model-based tracking and a 3D Hopfield neural network. *Machine Vision and Applications*, 10(5-6):292–307, 1998.

Guang-Bin Huang, Hongming Zhou, Xiaojian Ding, and Rui Zhang. Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 42(2):513–529, 2012.

Young-Seon Jeong, Myong K Jeong, and Olufemi A Omitaomu. Weighted dynamic time warping for time series classification. *Pattern Recognition*, 44(9):2231–2240, 2011.

Timor Kadir, Richard Bowden, Eng Jon Ong, and Andrew Zisserman. Minimal training, large lexicon, unconstrained sign language recognition. In *Proceedings of the British Machine Vision Conference*, volume 1, pages 1–10, 2004.

Harold W Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. In *Soviet Physics Doklady*, volume 10, page 707, 1966.

Jeroen F Lichtenauer, Emile A Hendriks, and Marcel JT Reinders. Sign language recognition by combining statistical DTW and independent classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):2040–2046, 2008.

Scott K Liddell and Robert E Johnson. American sign language: The phonological base. *Sign Language Studies*, 64:195–278, 1989.

Yui Man Lui. Human gesture recognition on product manifolds. *Journal of Machine Learning Research*, 13(1):3297–3321, 2012a.

Yui Man Lui. A least squares regression framework on manifolds and its application to gesture recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 13–18, 2012b.

Upal Mahbub, Tonmoy Roy, Md Shafiur Rahman, and Hafiz Imtiaz. One-shot-learning gesture recognition using motion history based gesture silhouettes. In *Proceedings of the International Conference on Industrial Application Engineering*, pages 186–193, 2013.

Manavender R Malgireddy, Ifeoma Inwogu, and Venu Govindaraju. A temporal Bayesian model for classifying, detecting and localizing activities in video sequences. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 43–48, 2012.

Manar Maraqa and Raed Abu-Zaiter. Recognition of arabic sign language (ArSL) using recurrent neural networks. In *Proceedings of the First International Conference on the Applications of Digital Information and Web Technologies*, pages 478–481, 2008.

Tin Hninn Hninn Maung. Real-time hand tracking and gesture recognition system using neural networks. *World Academy of Science, Engineering and Technology*, 50:466–470, 2009.

Sushmita Mitra and Tinku Acharya. Gesture recognition: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 37(3):311–324, 2007.

Kai Nickel and Rainer Stiefelhagen. Visual recognition of pointing gestures for human–robot interaction. *Image and Vision Computing*, 25(12):1875–1884, 2007.

Iason Oikonomidis, Nikolaos Kyriazis, and Antonis Argyros. Efficient model-based 3D tracking of hand articulations using Kinect. In *Proceedings of the British Machine Vision Conference*, pages 1–11, 2011.

Eng-Jon Ong and Richard Bowden. A boosted classifier tree for hand shape detection. In *Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 889–894, 2004.

Aditya Ramamoorthy, Namrata Vaswani, Santanu Chaudhury, and Subhashis Banerjee. Recognition of dynamic hand gestures. *Pattern Recognition*, 36(9):2069–2081, 2003.

Ingmar Rauschert, Pyush Agrawal, Rajeev Sharma, Sven Fuhrmann, Isaac Brewer, and Alan MacEachren. Designing a human-centered, multimodal GIS interface to support emergency management. In *Proceedings of the 10th ACM International Symposium on Advances in Geographic Information Systems*, pages 119–124, 2002.

Zhou Ren, Junsong Yuan, Jingjing Meng, and Zhengyou Zhang. Robust part-based hand gesture recognition using Kinect sensor. *IEEE Transactions on Multimedia*, 15(5):1110–1120, 2013.

Miguel Reyes, Gabriel Dominguez, and Sergio Escalera. Feature weighting in dynamic time warping for gesture recognition in depth data. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 1182–1188, 2011.

Yared Sabinas, Eduardo F Morales, and Hugo Jair Escalante. A One-Shot DTW-based method for early gesture recognition. In *Proceedings of 18th Iberoamerican Congress on Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pages 439–446, 2013.

Hae Jong Seo and Peyman Milanfar. Action recognition from one example. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):867–882, 2011.

Ling Shao and Ling Ji. Motion histogram analysis based key frame extraction for human action/activity representation. In *Proceedings of Canadian Conference on Computer and Robot Vision*, pages 88–92, 2009.

Jamie Shotton, Andrew W. Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. Real-time human pose recognition in parts from single depth images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1297–1304, 2011.

E Stergiopoulou and N Papamarkos. Hand gesture recognition using a neural network shape fitting technique. *Engineering Applications of Artificial Intelligence*, 22(8):1141–1158, 2009.

William C Stokoe. Sign language structure: An outline of the visual communication systems of the american deaf. *Studies in Linguistics, Occasional Papers*, 8, 1960.

Mu-Chun Su. A fuzzy rule-based approach to spatio-temporal hand gesture recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 30(2):276–281, 2000.

Christian Vogler and Dimitris Metaxas. Parallel hidden markov models for american sign language recognition. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 1, pages 116–122, 1999.

Christian Philipp Vogler. *American Sign Language Recognition: Reducing the Complexity of the Task with Phoneme-based Modeling and Parallel Hidden Markov Models*. PhD thesis, University of Pennsylvania, 2003.

J. Wachs, M. Kolsch, H. Stem, and Y. Edan. Vision-based hand-gesture applications. *Communications of the ACM*, 54(2):60–71, 2011.

Jun Wan, Qiuqi Ruan, Gaoyun An, and Wei Li. Gesture recognition based on hidden markov model from sparse representative observations. In *Proceedings of the IEEE 11th International Conference on Signal Processing*, volume 2, pages 1180–1183, 2012a.

Jun Wan, Qiuqi Ruan, Gaoyun An, and Wei Li. Hand tracking and segmentation via graph cuts and dynamic model in sign language videos. In *Proceedings of IEEE 11th International Conference on Signal Processing*, volume 2, pages 1135–1138. IEEE, 2012b.

Jun Wan, Qiuqi Ruan, Wei Li, and Shuang Deng. One-shot learning gesture recognition from RGB-D data using bag of features. *Journal of Machine Learning Research*, 14(1): 2549–2582, 2013.

Jun Wan, Vassilis Athitsos, Pat Jangyodsuk, Hugo Jair Escalante, Qiuqi Ruan, and Isabelle Guyon. CSMMI: Class-specific maximization of mutual information for action and gesture recognition. *IEEE Transactions on Image Processing*, 23(7):3152–3165, 2014a.

Jun Wan, Qiuqi Ruan, Wei Li, Gaoyun An, and Ruizhen Zhao. 3D SMoSIFT: Three-dimensional sparse motion scale invariant feature transform for activity recognition from RGB-D videos. *Journal of Electronic Imaging*, 23(2):023017, 2014b.

Chunli Wang, Wen Gao, and Shiguang Shan. An approach based on phonemes to large vocabulary Chinese sign language recognition. In *Proceedings of the IEEE Conference on Automatic Face and Gesture Recognition*, pages 411–416, 2002.

J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1290–1297, 2012.

Shu-Fai Wong, Tae-Kyun Kim, and Roberto Cipolla. Learning motion categories using both semantic and structural information. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–6, 2007.

Di Wu, Fan Zhu, and Ling Shao. One shot learning gesture recognition from RGBD images. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 7–12, 2012a.

Shen Wu, Feng Jiang, Debin Zhao, Shaohui Liu, and Wen Gao. Viewpoint-independent hand gesture recognition system. In *Proceedings of the IEEE Conference on Visual Communications and Image Processing*, pages 43–48, 2012b.

Morteza Zahedi, Daniel Keysers, and Hermann Ney. Appearance-based recognition of words in american sign language. In *Proceedings of Second Iberian Conference on Pattern Recognition and Image Analysis*, pages 511–519, 2005.

Liang-Guo Zhang, Yiqiang Chen, Gaolin Fang, Xilin Chen, and Wen Gao. A vision-based sign language recognition system using tied-mixture density HMM. In *Proceedings of the 6th International Conference on Multimodal Interfaces*, pages 198–204, 2004.

Journal of Machine Learning Research 16 (2015) 255-284 Submitted 3/14; Revised 9/14; Published 2/15

Multimodal Gesture Recognition via Multiple Hypotheses Rescoring

Vassilis Pitsikalis VPITSIK@CS.NTUA.GR
Athanasios Katsamanis NKATSAM@CS.NTUA.GR
Stavros Theodorakis STH@CS.NTUA.GR
Petros Maragos MARAGOS@CS.NTUA.GR
National Technical University of Athens
School of Electrical and Computer Engineering
Zografou Campus, Athens 15773, Greece

Editors: Isabelle Guyon, Vassilis Athitsos and Sergio Escalera

Abstract

We present a new framework for multimodal gesture recognition that is based on a multiple hypotheses rescoring fusion scheme. We specifically deal with a demanding Kinect-based multimodal data set, introduced in a recent gesture recognition challenge (ChaLearn 2013), where multiple subjects freely perform multimodal gestures. We employ multiple modalities, that is, visual cues, such as skeleton data, color and depth images, as well as audio, and we extract feature descriptors of the hands' movement, handshape, and audio spectral properties. Using a common hidden Markov model framework we build single-stream gesture models based on which we can generate multiple single stream-based hypotheses for an unknown gesture sequence. By multimodally rescoring these hypotheses via constrained decoding and a weighted combination scheme, we end up with a multimodally-selected best hypothesis. This is further refined by means of parallel fusion of the monomodal gesture models applied at a segmental level. In this setup, accurate gesture modeling is proven to be critical and is facilitated by an activity detection system that is also presented. The overall approach achieves 93.3% gesture recognition accuracy in the ChaLearn Kinect-based multimodal data set, significantly outperforming all recently published approaches on the same challenging multimodal gesture recognition task, providing a relative error rate reduction of at least 47.6%.

Keywords: multimodal gesture recognition, HMMs, speech recognition, multimodal fusion, activity detection

1. Introduction

Human communication and interaction takes advantage of multiple sensory inputs in an impressive way. Despite receiving a significant flow of multimodal signals, especially in the audio and visual modalities, our cross-modal integration ability enables us to effectively perceive the world around us. Examples span a great deal of cases. Cross-modal illusions are indicative of lower perceptual multimodal interaction and plasticity (Shimojo and Shams, 2001): for instance, when watching a video, a sound is perceived as coming from the speakers lips (the ventriloquism effect) while, in addition, speech perception may be affected by whether the lips are visible or not (the McGurk effect).

At a higher level, multimodal integration is also regarded important for language production and this is how the notion of multimodal gestures can be introduced. Several authors, as McNeill (1992), support the position that hand gestures hold a major role, and together with speech they are considered to have a deep relationship and to form an integrated system (Bernardis and Gentilucci, 2006) by interacting at multiple linguistic levels. This integration has been recently explored in terms of communication by means of language comprehension (Kelly et al., 2010). For instance, speakers pronounce words while executing hand gestures that may have redundant or complementary nature, and even blind speakers gesture while talking to blind listeners (Iverson and Goldin-Meadow, 1998). From a developmental point of view, see references in the work of Bernardis and Gentilucci (2006), hand movements occur in parallel during babbling of 6-8 month children, whereas word comprehension at the age of 8-10 months goes together with deictic gestures. All the above suffice to provide indicative evidence from various perspectives that hand gestures and speech seem to be interwoven.

In the area of human-computer interaction gesture has been gaining increasing attention (Turk, 2014). This is attributed both to recent technological advances, such as the wide spread of depth sensors, and to groundbreaking research since the famous "put that there" (Bolt, 1980). The natural feeling of gesture interaction can be significantly enhanced by the availability of multiple modalities. Static and dynamic gestures, the form of the hand, as well as speech, all together compose an appealing set of modalities that offers significant advantages (Oviatt and Cohen, 2000).

In this context, we focus on the effective detection and recognition of multimodally expressed gestures as performed *freely* by multiple users. Multimodal gesture recognition (MGR) poses numerous challenging research issues, such as detection of meaningful information in audio and visual signals, extraction of appropriate features, building of effective classifiers, and multimodal combination of multiple information sources (Jaimes and Sebe, 2007). The demanding data set (Escalera et al., 2013b) used in our work has been recently acquired for the needs of the multimodal gesture recognition challenge (Escalera et al., 2013a). It comprises multimodal cultural-anthropological gestures of everyday life, in spontaneous realizations of both spoken and hand-gesture articulations by multiple subjects, intermixed with other random and irrelevant hand, body movements and spoken phrases.

A successful multimodal gesture recognition system is expected to exploit both speech and computer vision technologies. Speech technologies and automatic speech recognition (Rabiner and Juang, 1993) have a long history of advancements and can be considered mature when compared to the research challenges found in corresponding computer vision tasks. The latter range from low-level tasks that deal with visual descriptor representations (Li and Allinson, 2008), to more difficult ones, such as recognition of action (Laptev et al., 2008), of facial expressions, handshapes and gestures, and reach higher-level tasks such as sign language recognition (Agris et al., 2008). However, recently the incorporation of depth enabled sensors has assisted to partially overcome the burden of detection and tracking, opening the way for addressing more challenging problems. The study of *multiple modalities' fusion* is one such case, that is linked with subjects discussed above.

Despite the progress seen in either unimodal cases such as the fusion of multiple speech cues for speech recognition (e.g., Bourlard and Dupont, 1997) or the multimodal case of audio-visual speech (Potamianos et al., 2004; Glotin et al., 2001; Papandreou et al., 2009),

the integration of dissimilar cues in MGR poses several challenges; even when several cues are excluded such as facial ones, or the eye gaze. This is due to the complexity of the task that involves several intra-modality diverse cues, as the 3D hands' shape and pose. These require different representations and may occur both sequentially and in parallel, and at different time scales and/or rates. Most of the existing gesture-based systems have certain limitations, for instance, either by only allowing a reduced set of symbolic commands based on simple hand postures or 3D pointing (Jaimes and Sebe, 2007), or by considering single-handed cases in controlled tasks. Such restrictions are indicative of the task's difficulty despite already existing work (Sharma et al., 2003) even before the appearance of depth sensors (Weimer and Ganapathy, 1989).

The fusion of multiple information sources can be either early, late or intermediate, that is, either at the data/feature level, or at the stage of decisions after applying independent unimodal models, or in-between; for further details refer to relative reviews (Jaimes and Sebe, 2007; Maragos et al., 2008). In the case of MGR late fusion is a typical choice since involved modalities may demonstrate synchronization in several ways (Habets et al., 2011) and possibly at higher linguistic levels. This is in contrast, for instance, to the case of combining lip movements with speech in audio-visual speech where early or state-synchronous fusion can be applied, with synchronization at the phoneme-level.

In this paper, we present a multimodal gesture recognition system that exploits the color, depth and audio signals captured by a Kinect sensor. The system first extracts features for the handshape configuration, the movement of the hands and the speech signal. Based on the extracted features and statistically trained models, single modality-based hypotheses are then generated for an unknown gesture sequence. The underlying single-modality modeling scheme is based on gesture-level hidden Markov models (HMMs), as described in Section 3.1. These are accurately initialized by means of a model-based activity detection system for each modality, presented in Section 3.3. The generated hypotheses are re-evaluated using a statistical *multimodal multiple hypotheses fusion* scheme, presented in Section 3.2. The proposed scheme builds on previous work on N-best rescoring: N-best sentence hypotheses scoring was introduced for the integration of speech and natural language by Chow and Schwartz (1989) and has also been employed for the integration of different recognition systems based on the same modality, e.g., by Ostendorf et al. (1991), or for audio-visual speech recognition by Glotin et al. (2001). *Given* the best multimodally-selected hypothesis, and the implied gesture temporal boundaries in all information streams, a final *segmental parallel fusion* step is applied based on parallel HMMs (Vogler and Metaxas, 2001). We show in Section 5 that the proposed overall MGR framework outperforms the approaches that participated in the recent demanding multimodal challenge (Escalera et al., 2013a), as published in the proceedings of the workshop, by reaching an accuracy of 93.3 and leading to a relative error rate (as Levenshtein distance) reduction of 47% over the first-ranked team.

2. Related Work

Despite earlier work in multimodal gesture recognition, it is considered an open field, related to speech recognition, computer vision, gesture recognition and human-computer interaction. As discussed in Section 1 it is a multilevel problem posing challenges on audio

and visual processing, on multimodal stream modeling and fusion. Next, we first consider works related to the recent advances on multimodal recognition, including indicative works evaluated in the same ChaLearn challenge and recognition task by sharing the exact training/testing protocol and data set. Then, we review issues related to basic components and tasks, such as visual detection and tracking, visual representations, temporal segmentation, statistical modeling and fusion.

There are several excellent reviews on multimodal interaction either from the computer vision or human-computer interaction aspect (Jaimes and Sebe, 2007; Turk, 2014). Since earlier pioneering works (Bolt, 1980; Poddar et al., 1998) there has been an explosion of works in the area; this is also due to the introduction of everyday usage depth sensors (e.g., Ren et al., 2011). Such works span a variety of applications such as the recent case of gestures and accompanying speech integration for a problem in geometry (Miki et al., 2014), the integration of nonverbal auditory features with gestures for agreement recognition (Bousmalis et al., 2011), or within the aspect of social signal analysis (Ponce-López et al., 2013); Song et al. (2013) propose a probabilistic extension of first-order logic, integrating multimodal speech/visual data for recognizing complex events such as everyday kitchen activities.

The ChaLearn task is an indicative case of the effort recently placed in the field: Published approaches ranked in the first places of this gesture challenge, employ multimodal signals including audio, color, depth and skeletal information; for learning and recognition one finds approaches ranging from hidden Markov models (HMMs)/Gaussian mixture models (GMMs) to boosting, random forests, neural networks and support vector machines among others. Next, we refer to indicative approaches from therein, (Escalera et al., 2013b). In Section 5 we refer to specific details for the top-ranked approaches that we compare with. Wu et al. (2013), the first-ranked team, are driven by the audio modality based on end-point detection, to detect the multimodal gestures; then they combine classifiers by calculating normalized confidence scores. Bayer and Thierry (2013) are also driven by the audio based on a hand-tuned detection algorithm, then they estimate class probabilities per gesture segment and compute their weighted average. Nandakumar et al. (2013) are driven by both audio HMM segmentation, and skeletal points. They discard segments not detected in both modalities while employing a temporal overlap coefficient to merge overlapping modalities' segments. Finally, they recognize the gesture with the highest combined score. Chen and Koskela (2013) employ the extreme learning machine, a class of single-hidden layer feed-forward neural network and apply both early and late fusion. In a late stage, they use the geometric mean to fuse the classification outputs. Finally, Neverova et al. (2013) propose a multiple-scale learning approach that is applied on both temporal and spatial dimension while employing a recurrent neural network. Our contribution in the specific area of multimodal gestures recognition concerns the employment of a late fusion scheme based on multiple hypothesis rescoring. The proposed system, also employing multimodal activity detectors, all in a HMM statistical framework, demonstrates improved performance over the rest of the approaches that took part in the specific ChaLearn task.

From the visual processing aspect the first issue to be faced is *hand detection* and *tracking*. Regardless of the boost offered after the introduction of depth sensors there are unhandled cases as in the case of low quality video or resolution, in complex scene backgrounds with multiple users, and varying illumination conditions. Features employed are related to skin color, edge information, shape and motion for hand detection (Argyros

and Lourakis, 2004; Yang et al., 2002), and learning algorithms such as boosting (Ong and Bowden, 2004). *Tracking* is based on blobs (Starner et al., 1998; Tanibata et al., 2002; Argyros and Lourakis, 2004), hand appearance (Huang and Jeng, 2001), or hand boundaries (Chen et al., 2003; Cui and Weng, 2000), whereas modeling techniques include Kalman filtering (Binh et al., 2005), the condensation method (Isard and Blake, 1998), or full upper body pose tracking (Shotton et al., 2013). Others directly employ global image features (Bobick and Davis, 2001). Finally, Alon et al. (2009) employ a unified framework that performs spatial segmentation simultaneously with higher level tasks. In this work, similarly to other authors, see works presented by Escalera et al. (2013b), we take advantage of the Kinect-provided skeleton tracking.

Visual feature extraction aims at the representation of the movement, the position and the shape of the hands. Representative measurements include the center-of-gravity of the hand blob (Bauer and Kraiss, 2001), motion features (Yang et al., 2002), as well as features related with the hand's shape, such as shape moments (Starner et al., 1998) or sizes and distances within the hand (Vogler and Metaxas, 2001). The contour of the hand is also used for invariant features, such as Fourier descriptors (Conseil et al., 2007). handshape representations are extracted via principal component analysis (e.g., Du and Piater, 2010), or with variants of active shape and appearance models (Roussos et al., 2013). Other approaches (e.g. Dalal and Triggs, 2005) employ general purpose features as the Histogram of Oriented Gradients (HOG) (Buehler et al., 2009), or the scale invariant feature transform (Lowe, 1999). Li and Allinson (2008) present a review on local features. In this work, we employ the 3D points of the articulators as extracted from the depth-based skeleton tracking and the HOG descriptors for the handshape cue.

Temporal detection or *segmentation* of meaningful information concerns another important aspect of our approach. Often the segmentation problem is seen in terms of gesture spotting, that is, for the detection of the meaningful gestures, as adapted from the case of speech (Wilcox and Bush, 1992) where all non-interesting patterns are modeled by a single filler model. Specifically, Lee and Kim (1999) employ in similar way an ergodic model termed as threshold model to set adaptive likelihood thresholds. Segmentation may be also seen in combination with recognition as by Alon et al. (2009) or Li and Allinson (2007); in the latter, start and end points of gestures are determined by zero crossing of likelihoods' difference between gesture/non-gestures. There has also been substantial related work in sign language tasks: Han et al. (2009) explicitly perform segmentation based on motion discontinuities, Kong and Ranganath (2010) segment trajectories via rule-based segmentation, whereas others apply systematic segmentation as part of the modeling of sub-sign components (sub-units) (Bauer and Kraiss, 2001); the latter can be enhanced by an unsupervised segmentation component (Theodorakis et al., 2014) or by employing linguistic-phonetic information (Pitsikalis et al., 2011), leading to multiple subunit types. In our case, regardless of the availability of ground truth temporal gesture annotations we employ independent monomodal model-based activity detectors that share a common HMM framework. These function independently of the ground truth annotations, and are next exploited at the statistical modeling stage.

Multimodal gesture recognition concerns multiple dynamically varying streams, requiring the handling of multiple variable time-duration diverse cues. Such requirements are met by approaches such as hidden Markov models that have been found to efficiently model

temporal information. The corresponding framework further provides efficient algorithms, such as BaumWelch and Viterbi (Rabiner and Juang, 1993), for evaluation, learning, and decoding. For instance, Nam and Wohn (1996) apply HMMs in gesture recognition, Lee and Kim (1999) in gesture spotting, whereas parametric HMMs (Wilson and Bobick, 1999) are employed for gestures with systematic variation. At the same time parallel HMMs (Vogler and Metaxas, 2001) accommodate multiple cues simultaneously. Extensions include conditional random fields (CRFs) or generalizations (Wang et al., 2006), while non-parametric methods are also present in MGR tasks (Celebi et al., 2013; Hernández-Vela et al., 2013). In this paper we build word-level HMMs, which fit our overall statistical framework, both for audio and visual modalities, while also employing parallel HMMs for late fusion.

3. Proposed Methodology

To better explain the proposed multimodal gesture recognition framework let us first describe a use case. Multimodal gestures are commonly used in various settings and cultures (Morris et al., 1979; Kendon, 2004). Examples include the "OK" gesture expressed by creating a circle using the thumb and forefinger and holding the other fingers straight and at the same time uttering "Okay" or "Perfect". Similarly, the gesture "Come here" involves the generation of the so-called beckoning sign which in Northern America is made by sticking out and moving repeatedly the index finger from the clenched palm, facing the gesturer, and uttering a phrase such as "Come here" or "Here". We specifically address automatic detection and recognition of a set of such spontaneously generated multimodal gestures even when these are intermixed with other irrelevant actions, which could be verbal, nonverbal or both. The gesturer may, for example, be walking in-between the gestures or talking to somebody else.

In this context, we focus only on gestures that are always multimodal, that is, they are not expressed only verbally or non-verbally, without implying however strictly synchronous realizations in all modalities or making any related assumptions apart from expecting consecutive multimodal gestures to be sufficiently well separated in time, namely a few milliseconds apart in all information streams. Further, no linguistic assumptions are made regarding the sequence of gestures, namely any gesture can follow any other.

Let $V_g = \{g_i\}, i = 1, \ldots, |V_g|$ be the vocabulary of multimodal gestures g_i that are to be detected and recognized in a recording and let $S = \{\mathbf{O}_i\}, i = 1, \ldots, |S|$ be the set of information streams that are concurrently observed for that purpose. In our experiments, the latter set comprises three streams, namely audio spectral features, the gesturer's skeleton and handshape features. Based on these observations the proposed system will generate a hypothesis for the sequence of gesture appearances in a specific recording/session, like the following:

$$\mathbf{h} = [bm, g_1, sil, g_5, \ldots, bm, sil, g_3].$$

The symbol *sil* essentially corresponds to inactivity in all modalities while *bm* represents any other activity, mono- or multimodal, that does not constitute any of the target multimodal gestures. This recognized sequence is generated by exploiting single stream-based gesture models via the proposed fusion algorithm that is summarized in Figure 1 and described in detail in Section 3.2. For the sake of clarity, the single stream modeling framework is first presented in Section 3.1. Performance of the overall algorithm is found to depend on how

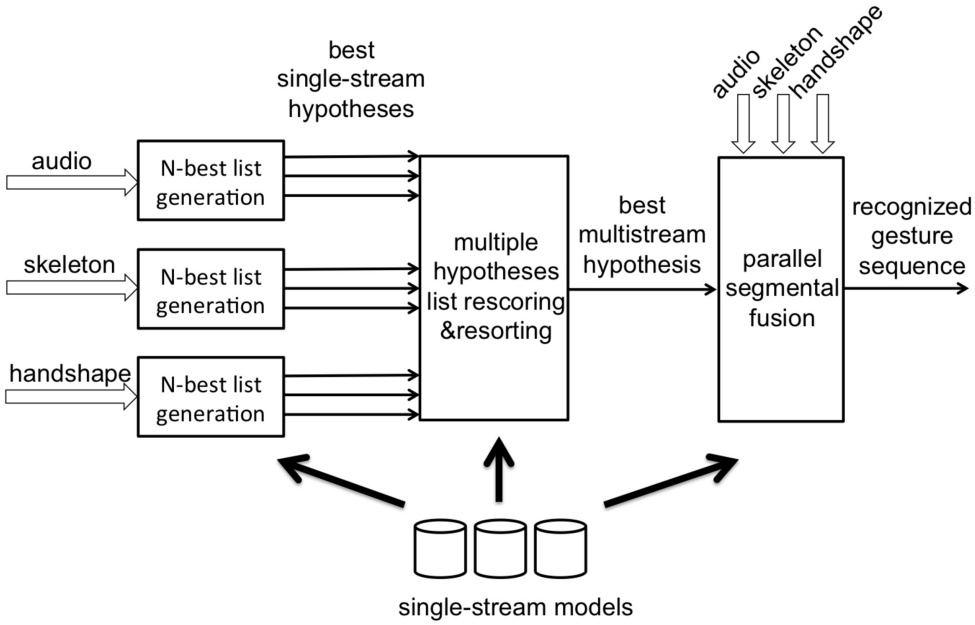


Figure 1: Overview of the proposed multimodal fusion scheme for gesture recognition based on multiple hypotheses rescoring. Single-stream models are first used to generate possible hypotheses for the observed gesture sequence. The hypotheses are then rescored by all streams and the best one is selected. Finally, the observed sequence is segmented at the temporal boundaries suggested by the selected hypothesis and parallel fusion is applied to classify the resulting segments. Details are given in Section 3.2.

accurately the single stream models represent each gesture. This representation accuracy can be significantly improved by the application of the multimodal activity detection scheme described in Section 3.3.

3.1 Speech, Skeleton and Handshape Modeling

The underlying single-stream modeling scheme is based on Hidden Markov Models (HMMs) and builds on the keyword-filler paradigm that was originally introduced for speech (Wilpon et al., 1990; Rose and Paul, 1990) in applications like spoken document indexing and retrieval (Foote, 1999) or speech surveillance (Rose, 1992). The problem of recognizing a limited number of gestures in an observed sequence comprising other heterogeneous events as well, is seen as a keyword detection problem. The gestures to be recognized are the keywords and all the rest is ignored. Then, for every information stream, each gesture $g_i \in V_g$, or, in practice, its projection on that stream, is modeled by an HMM and there are two separate filler HMMs to represent either silence/inactivity (sil) or all other possible events (bm) appearing in that stream.

All these models are basically left-to-right HMMs with Gaussian mixture models (GMMs) representing the state-dependent observation probability distributions. They are initialized by an iterative procedure which sets the model parameters to the mean and covariance of the features in state-corresponding segments of the training instances and refines the segment boundaries via the Viterbi algorithm (Young et al., 2002). Training is performed using the Baum-Welch algorithm (Rabiner and Juang, 1993), and mixture components are incrementally refined.

While this is the general training procedure followed, two alternative approaches are investigated, regarding the exact definition and the supervised training process of all involved models. These are described in the following. We experiment with both approaches and we show that increased modeling accuracy at the single-stream level leads to better results overall.

3.1.1 Training Without Employing Activity Detection

In this case, single-stream models are initialized and trained based on coarse, multimodal temporal annotations of the gestures. These annotations are common for all streams and given that there is no absolute synchronization across modalities they may also include inactivity or other irrelevant events in the beginning or end of the target gestural expression. In this way the gesture models already include, by default, inactivity segments. As a consequence we do not train any separate inactivity (*sil*) model. At the same time, the background model (*bm*) is trained on all training instances of all the gestures, capturing in this way only generic gesture properties that are expected to characterize a non-target gesture. The advantage of this approach is that it may inherently capture cross-modal synchronicity relationships. For example, the waving hand motion may start before speech in the waving gesture and so there is probably some silence (or other events) to be expected before the utterance of a multimodal gesture (e.g. "Bye bye") which is modeled implicitly.

3.1.2 Training With Activity Detection

On the other hand, training of single-stream models can be performed completely independently using stream-specific temporal boundaries of the target expressions. In this direction, we applied an activity detection scheme, described in detail in Section 3.3. Based on that, it is possible to obtain tighter stream-specific boundaries for each gesture. Gesture models are now trained using these tighter boundaries, the *sil* model is trained on segments of inactivity (different for each modality) and the *bm* model is trained on segments of activity but outside the target areas. In this case, single-stream gesture models can be more accurate but any possible evidence regarding synchronicity across modalities is lost.

3.2 Multimodal Fusion of Speech, Skeleton and Handshape

Using the single-stream gesture models (see Section 3.1) and a gesture-loop grammar as shown in Figure 2(a) we initially generate a list of N-best possible hypotheses for the unknown gesture sequence for each stream. Specifically, the Viterbi algorithm (Rabiner and Juang, 1993) is used to directly estimate the best stream-based possible hypothesis $\hat{\mathbf{h}}_m$

Algorithm 1 Multimodal Scoring and Resorting of Hypotheses

```
% N-best list rescoring
for all hypotheses do
    % Create a constrained grammar
    keep the sequence of gestures fixed
    allow insertion/deletion of sil and bm occurrences between gestures
    for all modalities do
        by applying the constrained grammar and Viterbi decoding:
        1) find the best state sequence given the observations
        2) save corresponding score and temporal boundaries
    % Late fusion to rescore hypotheses
    final hypothesis score is a weighted sum of modality-based scores
the best hypothesis of the 1st-pass is the one with the maximum score
```

for the unknown gesture sequence as follows:

$$\hat{\mathbf{h}}_m = \arg\max_{\mathbf{h}_m \in G} \log P(\mathbf{O}_m | \mathbf{h}_m, \lambda_m), \quad m = 1, \ldots, |S|$$

where $\mathbf{O}_m$ is the observation[1] sequence for modality m, λ_m is the corresponding set of models and G is the set of alternative hypotheses allowed by the gesture loop grammar. Instead of keeping just the best scoring sequence we apply essentially a variation of the Viterbi algorithm, namely the lattice N-best algorithm (Shwartz and Austin, 1991), that apart from storing just the single best gesture at each node it also records additional best-scoring gestures together with their scores. Based on these records, a list of N-best hypotheses for the entire recording and for each modality can finally be estimated.

The N-best lists are generated independently for each stream and the final superset of the multimodally generated hypotheses may contain multiple instances of the same gesture sequence. By removing possible duplicates we end up with L hypotheses forming the set $H = \{\mathbf{h}_1, \ldots, \mathbf{h}_L\}$; $\mathbf{h}_i$ is a gesture sequence (possibly including *sil* and *bm* occurrences as well). Our goal is to sort this set and identify the most likely hypothesis this time exploiting all modalities together.

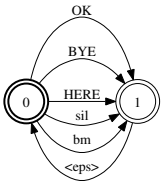
3.2.1 Multimodal Scoring and Resorting of Hypotheses

In this direction, and as summarized in Algorithm 1, we estimate a combined score for each possible gesture sequence as a weighted sum of modality-based scores

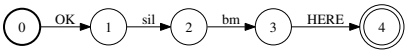
$$v_i = \sum_{m \in S} w_m v_{m,i}^s, \quad i = 1 \ldots L, \tag{1}$$

where the weights w_m are determined experimentally in a left-out validation set of multimodal recordings. The validation set is distinct from the final evaluation (test) set; more

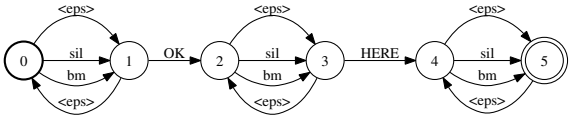
1. For the case of video data an observation corresponds to a single image frame; for the case of audio modality it corresponds to a 25 msec window.



(a) gesture-loop



(b) hypothesis



(c) hypothesis-dependent grammar

Figure 2: Finite-state-automaton (FSA) representations of finite state grammars: (a) an example gesture-loop grammar with 3 gestures plus inactivity and background labels. The "eps" transition represents an ϵ transition of the FSA, (b) an example hypothesis, (c) a hypothesis-dependent grammar allowing varying *sil* and *bm* occurrences between gestures.

Algorithm 2 Segmental Parallel Fusion

```
% Parallel scoring
for all modalities do segment observations based on given temporal boundaries
    for all resulting segments do
        estimate a score for each gesture given the segment observations
        temporally align modality segments
        for all aligned segments do
            estimate weighted sum of modality-based scores for all gestures
            select the best-scoring gesture (sil and bm included)
```

details on the selection of weights are provided in Section 5. The modality-based scores $v^s_{m,i}$ are standardized versions[2] of $v_{m,i}$ which are estimated by means of Viterbi decoding as follows:

$$v_{m,i} = \max_{\mathbf{h} \in G_{h_i}} \log P(\mathbf{O}_m | \mathbf{h}, \lambda_m), \quad i = 1, \ldots, L, \ m = 1, \ldots, |S| \tag{2}$$

where $\mathbf{O}_m$ is the observation sequence for modality m and λ_m is the corresponding set of models. This actually solves a constrained recognition problem in which acceptable gesture sequences need to follow a specific hypothesis-dependent finite state grammar G_{h_i}. It is required that the search space of possible state sequences only includes sequences

2. That is, transformed to have zero mean and a standard deviation of one.

corresponding to the hypothesis $\mathbf{h}_i$ plus possible variations by keeping the appearances of target gestures unaltered and only allow *sil* and *bm* labels to be inserted, deleted and substituted with each other. An example of a hypothesis and the corresponding grammar is shown in Figure 2(b,c). In this way, the scoring scheme accounts for inactivity or non-targeted activity that is not necessarily multimodal, e.g., the gesturer is standing still but speaking or is walking silently. This is shown to lead to additional improvements when compared to a simple forced-alignment based approach.

It should be mentioned that hypothesis scoring via (2) can be skipped for the modalities based on which the particular hypothesis was originally generated. These scores are already available from the initial N-best list estimation described earlier.

The best hypothesis at this stage is the one with the maximum combined score as estimated by (1). Together with the corresponding temporal boundaries of the included gesture occurrences, which can be different for the involved modalities, this hypothesized gesture sequence is passed on to the segmental parallel scoring stage. At this last stage, only local refinements are allowed by exploiting possible benefits of a segmental classification process.

3.2.2 Segmental Parallel Fusion

The segmental parallel fusion algorithm is summarized in Algorithm 2. Herein we exploit the modality-specific time boundaries for the most likely gesture sequence determined in the previous step, to reduce the recognition problem into a segmental classification one. First, we segment the audio, skeleton and handshape observation streams employing these boundaries. Given that in-between gestures, i.e., for *sil* or *bm* parts, there may not be one-to-one correspondence between segments of different observation streams these segments are first aligned with each other across modalities by performing an optimal symbolic string match using dynamic programming. Then, for every aligned segment t and each information stream m we compute the log probability

$$LL^t_{m,j} = \max_{\mathbf{q}\in Q} \log P(\mathbf{O}^t_m, \mathbf{q}|\lambda_{m,j}), \quad j = 1, \ldots, |V_g| + 2,$$

where $\lambda_{m,j}$ are the parameters of the model for the gesture g_j in the extended vocabulary $V_g \cup \{sil, bm\}$ and the stream $m \in S$; $\mathbf{q}$ is a possible state ($\in Q$) sequence. These segmental scores are linearly combined across modalities to get a multimodal gestural score (left hand side) for each segment

$$LL^t_j = \sum_{m\in S} w'_m LL^t_{m,j}, \tag{3}$$

where w'_m, is the stream-weight for modality m set to optimize recognition performance in a validation data set.[3] Finally, the gesture with the highest score is the recognized one for each segment t. This final stage is expected to give additional improvements and correct false alarms by seeking loosely overlapping multimodal evidence in support of each hypothesized gesture.

3. The w'_m are different from the weights in (1). Their selection is similarly based on a separate validation set that is distinct from the final evaluation set; more details on the selection of weights are provided in Section 5.

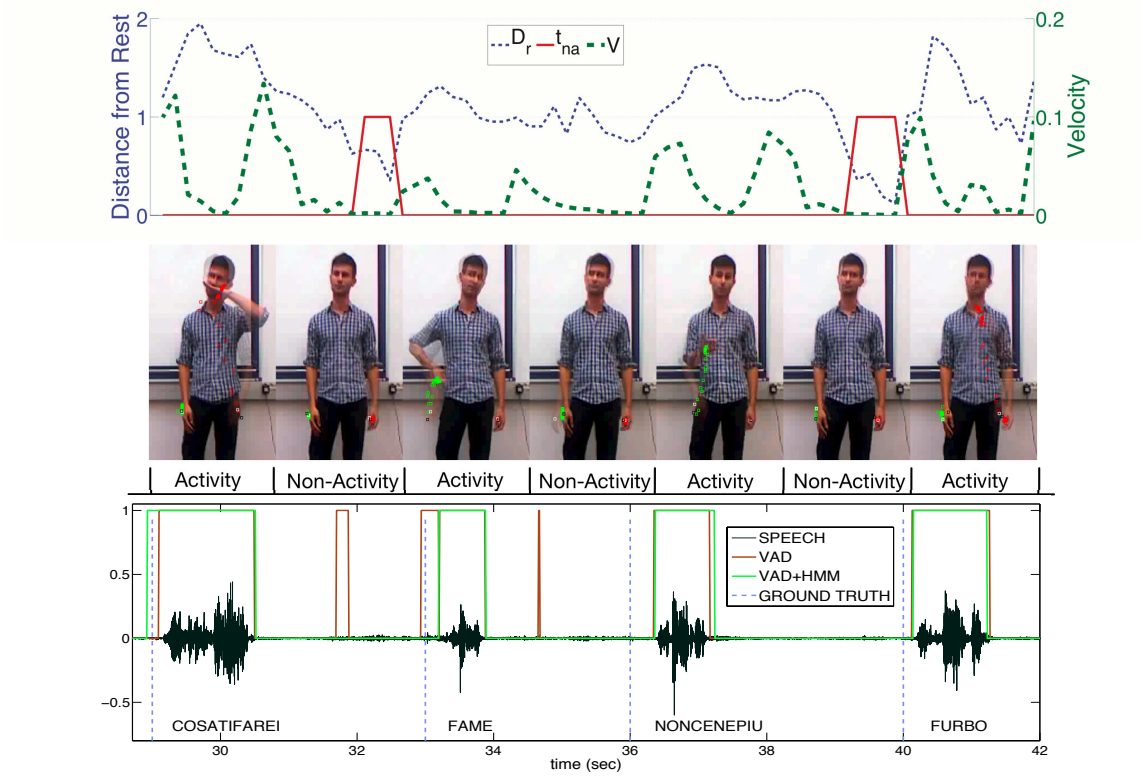


Figure 3: Activity detection example for both audio and visual modalities for one utterance. First row: The velocity of the hands (V), their distance with respect to the rest position (D_r) and the resulting initial estimation of gesture non-activity segments (t_{na}). Second row: The estimated gesture activity depicted on the actual video images. Third row: The speech signal accompanied with the initial VAD, the VAD+HMM and the gesture-level temporal boundaries included in the gesture data set (ground truth).

3.3 Multimodal Activity Detection

To achieve activity detection for each one of visual and audio modalities, we follow a common model-based framework. This is based on two complementary models of "activity" and "non-activity". In practice, these models, have different interpretations for the different modalities. This is first due to the nature of each modality, and second due to challenging data acquisition conditions. For the case of speech, the non-activity model may correspond to noisy conditions, e.g., keyboard typing or fan noise. For the case of the visual modality, the non-activity model refers to the rest cases in-between the articulation of gestures. However, these rests are not strictly defined, since the subject may not always perform a full rest and/or the hands may not stop moving. All cases of activity, in both the audio and the skeleton streams, such as out-of-vocabulary multimodal gestures and other

spontaneous gestures are thought to be represented by the activity model. Each modality's activity detector is initialized by a modality-specific front-end, as described in the following.

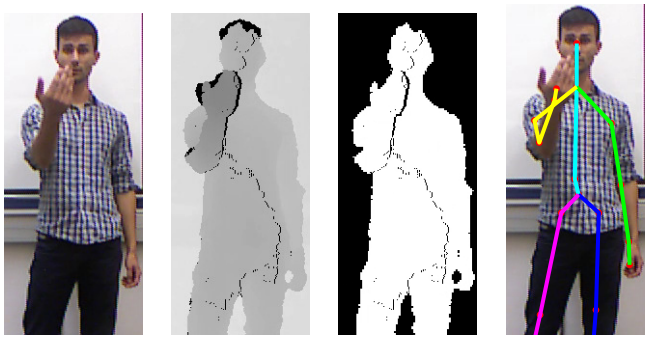
For the case of speech, activity and non-activity models are initialized on activity and non-activity segments correspondingly. These are determined by taking advantage for initialization of a Voice Activity Detection (VAD) method recently proposed by Tan et al. (2010). This method is based on likelihood ratio tests (LRTs) and by treating the LRT's for the voice/unvoiced frames differently it gives improved results than conventional LRT-based and standard VADs. The activity and non-activity HMM models are further trained using an iterative procedure employing the Baum-Welch algorithm, better known as embedded re-estimation (Young et al., 2002). The final boundaries of the speech activity and non-activity segments are determined by application of the Viterbi algorithm.

For the visual modality, the goal is to detect activity concerning the dynamic gesture movements versus the rest cases. For this purpose, we first initialize our non-activity models on rest position segments which are determined on a recording basis. For these segments skeleton movement is characterized by low velocity and the skeleton is close to the rest position $\mathbf{x_r}$. To identify non-active segments, we need to estimate a) the skeleton rest position b) the hands velocity, and c) the distance of the skeleton to that position. Hands' velocity is computed as $V(\mathbf{x}) = \|\dot{\mathbf{x}}\|$ where $\mathbf{x}(t)$ is the 3D hands' centroid coordinate vector and t is time. The rest position is estimated as the median skeleton position of all the segments for which hands' velocity V is below a certain threshold $V_{T_r} = 0.2 \cdot \bar{V}$, where $\bar{V}$ is the average velocity of all segments. The distance of the skeleton to the rest position is determined as $D_r(\mathbf{x}) = \|\mathbf{x} - \mathbf{x_r}\|$. Initial non-activity segments t_{na} are the ones for which the following two criteria hold, namely $\mathbf{t_{na}} = \{t : \ D_r(\mathbf{x}) < D_{T_r} and\ V(\mathbf{x}) < V_{T_r}\}$. Taking as input these t_{na} segments we train a non-activity HMM model while an activity model is trained on all remaining segments using the skeleton feature vector as described in Section 5.1.1. Further, similar to the case of speech we re-train the HMM models using embedded re-estimation. The final boundaries of the visual activity and non-activity segments are determined by application of the Viterbi algorithm.

In Figure 3, we illustrate an example of the activity detection for both audio and visual modalities for one utterance. In the first row, we depict the velocity of the hands (V), their distance with respect to the rest position (D_r) and the initial estimation of gesture non-activity (t_{na}) segments. We observe that in t_{na} segments both V and D_r are lower than the predefined thresholds ($V_{T_r} = 0.6, D_{T_r} = 0.006$)[4] and correspond to non-activity. In the second row, we illustrate the actual video frames images. These are marked with the tracking of both hands and accompanied with the final model-based gesture activity detection. In the bottom, we show the speech signal, with the initial VAD boundaries, the refined, HMM-based ones (VAD+HMM) and the gesture-level boundaries included in the data set (ground truth). As observed the refined detection (VAD+HMM) is tighter and more precise compared to the initial VAD and the data set annotations.

To sum up, after applying the activity detectors for both audio and visual modalities we merge the corresponding outputs with the gesture-level data set annotations in order to obtain refined stream-specific boundaries that align to the actual activities. In this way,

4. These parameters are set after experimentation in a single video of the validation set, that was annotated in terms of activity.



(a) RGB (b) Depth (c) Mask (d) Skeleton

Figure 4: Sample cues of the multimodal gesture challenge 2013 data set.

we compensate for the fact that the data set annotations may contain non-activity at the start/end of each gesture.

4. Multimodal Gestures' Data set

For our experiments we employ the ChaLearn multimodal gesture challenge data set, introduced by Escalera et al. (2013b). Other similar data sets are described by Ruffieux et al. (2013, 2014). This data set focuses on multiple instance, user independent learning of gestures from multi-modal data. It provides via Kinect RGB and depth images of face and body, user masks, skeleton information, joint orientation as well as concurrently recorded audio including the speech utterance accompanying/describing the gesture (see Figure 4). The vocabulary contains 20 Italian cultural-anthropological gestures. The data set contains three separate sets, namely for development, validation and final evaluation, including 39 users and 13858 gesture-word instances in total. All instances have been manually transcribed and loosely end-pointed. The corresponding temporal boundaries are also provided; these temporal boundaries are employed during the training phase of our system.

There are several issues that render multimodal gesture recognition in this data set quite challenging as described by Escalera et al. (2013b), such as the recording of continuous sequences, the presence of distracter gestures, the relatively large number of categories, the length of the gesture sequences, and the variety of users. Further, there is no single way to perform the included cultural gestures, e.g., "vieni qui" is performed with repeated movements of the hand towards the user, with a variable number of repetitions (see Figure 5). Similarly, single-handed gestures may be performed with either the left or right hand. Finally, variations in background, lighting and resolution, occluded body parts and spoken dialects have also been introduced.

5. Experiments

We first provide information on the multimodal statistical modeling that includes feature extraction and training. Then, we discuss the involved fusion parameters, the evaluation procedure, and finally, present results and comparisons.

(a) (b) (c) (d)

Figure 5: (a,b) Arm position variation (low, high) for gesture 'vieni qui'; (c,d) Left and right handed instances of 'vattene'.

5.1 Parameters, Evaluation, Structure

Herein, we describe first the employed feature representations, and training parameters for each modality, such as number of states and mixture components: as discussed in Section 3.1 we statistically train separate gesture HMMs per each information stream: skeleton, handshape and audio. Next, we describe the stream weight selection procedure, note the best stream weights, and present indicative results of the procedure. After presenting the evaluation metrics, we finally describe the overall rational of the experimental structure.

5.1.1 Multimodal Features, HMM and Fusion Parameters

The features employed for the *skeleton* cue include: the hands' and elbows' 3D position, the hands 3D velocity, the 3D direction of the hands' movement, and the 3D distance of hands' centroids. For the *handshape's* representation we employ the HOG feature descriptors. These are extracted on both hands' segmented images for both RGB and depth cues. We segment the hands by performing a threshold-based depth segmentation employing the hand's tracking information. For the *audio* modality we intend to efficiently capture the spectral properties of speech signals by estimating the Mel Frequency Cepstral Coefficients (MFCCs). Our front end generates 39 acoustic features every 10 msec. Each feature vector comprises 13 MFCCs along with their first and second derivatives. All the above feature descriptors are well known in the related literature. The specific selections should not affect the conclusions as related to the main fusion contributions, since these build on the level of the likelihoods. Such an example would be the employment of other descriptors as for instance in the case of visual (e.g., Li and Allinson, 2008) or speech related features (e.g., Hermansky, 1990).

For all modalities, we train separate gesture, *sil* and *bm* models as described in Section 3.1. These models are trained either using the data set annotations or based on the input provided by the activity detectors. The number of states, Gaussian components per state, stream weights and the word insertion penalty in all modalities are determined ex-

perimentally based on the recognition performance on the validation set.[5] For skeleton, we train left-right HMMs with 12 states and 2 Gaussians per state. For handshape, the models correspondingly have 8 states and 3 Gaussians per state while speech gesture models have 22 states and 10 Gaussians per state.

The training time is on average 1 minute per skeleton and handshape model and 90 minutes per audio model. The decoding time is on average 4xRT (RT refers to real-time).[6] A significant part of the decoding time is due to the generation of the N-best lists of hypotheses. In our experiments N is chosen to be equal to 200. We further observed that the audio-based hypotheses were always ranked higher than those from the other single-stream models. This motivated us to include only these hypotheses in the set we considered for rescoring.

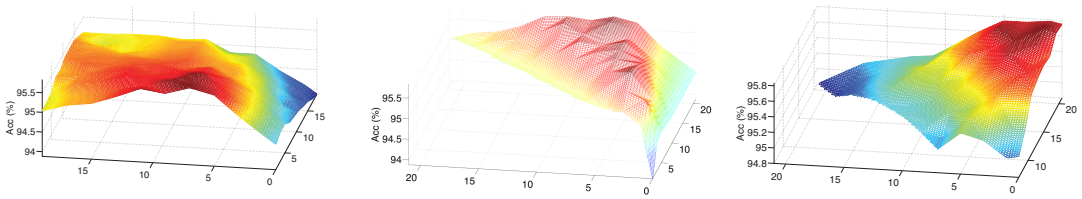
5.1.2 Stream Weight Configuration

Herein, we describe the experimental procedure for the selection of the stream weights $w_m, w'_m, m \in S$ of (1) and (3), for the components of multimodal hypothesis rescoring (MHS) and segmental parallel fusion (SPF). The final weight value selection is based on the optimization of recognition performance in the *validation* data set which is completely distinct from the final evaluation (test) data set.

Specifically, the w_m's are first selected from a set of alternative combinations to optimize gesture accuracy at the output of the MHS component. The SPF weights w'_m's are subsequently set to optimize the performance of the overall framework. The best weight combination for the multimodal hypothesis rescoring component is found to be $w^*_{SK,HS,AU} = [63.6, 9.1, 27.3]$, where SK, HS and AU correspond to skeleton, handshape and audio respectively.[7] This leads to the best possible accuracy of MHS in the validation set, namely 95.84%. Correspondingly, the best combination of weights for the segmental fusion component is $[0.6, 0.6, 98.8]$. Overall, the best achieved gesture recognition accuracy is 96.76% in the validation set.

In Figures 6(a), (b) and (c) we show the recognition accuracy of the MHS component for the various combinations of the w_m's. For visualization purposes we show accuracy when the weights vary in pairs and the remaining weight is set to its optimal value. For example, Figure 6(a) shows recognition accuracy for various combinations of handshape and audio weights when the skeleton weight is equal to 63.6. Overall, we should comment that the skeleton's contribution appears to be the most significant in the rescoring phase. This is of course a first interpretation, since the list of original hypotheses is already audio-based only, and the audio contribution cannot be directly inferred. As a consequence these results should be seen under this viewpoint. In any case, given that audio-based recognition leads to 94.1% recognition accuracy (in the validation set) it appears that both skeleton

5. Parameter ranges in the experiments for each modality are as follows. Audio: States 10-28, Gaussians: 2-32; Skeleton/handshape: States 7-15, Gaussians: 2-10.
6. For the measurements we employed an AMD Opteron(tm) Processor 6386 at 2.80GHz with 32GB RAM.
7. The weights take values in $[0, 1]$ while their sum across the modalities adds to one; these values are then scaled by 100 for the sake of numerical presentation. For the w stream weights we sampled the $[0, 1]$ with 12 samples for each modality, resulting to 1728 combinations. For the w' case, we sampled the $[0, 1]$ space by employing 5, 5 and 21 samples for the gesture, handshape and speech modalities respectively, resulting on 525 combinations.



(a) HS-AU (b) SK-AU (c) SK-HS

Figure 6: Gesture recognition accuracy of the Multiple hypothesis rescoring component for various weight-pair combinations. From left to right, the handshape-audio, skeleton-audio, skeleton-handshape weight pairs are varied. The remaining weight is set to its optimal value, namely 63.6 for skeleton, 9.1 for handshape and 27.3 for audio.

and handshape contribute in properly reranking the hypotheses and improve performance (which is again confirmed by the results in the test set presented in the following sections).

5.1.3 Evaluation

The presented evaluation metrics include the Levenshtein distance (LD)[8] which is employed in the ChaLearn publications (Escalera et al., 2013b) and the gesture recognition accuracy. The Levenshtein distance $LD(R,T)$, also known as "edit distance', is the minimum number of edit operations that one has to perform to go from symbol sequence R to T, or vice versa; edit operations include substitutions (S), insertions (I), or deletions (D). The overall score is the sum of the Levenshtein distances for all instances compared to the corresponding ground truth instances, divided by the total number of gestures. At the same time we report the standard word recognition accuracy $Acc = 1 - LD = \frac{N-S-D-I}{N}$, where N is the total number of instances of words.

Finally, we emphasize that all reported results have been generated by strictly following the original ChaLearn challenge protocol which means that they are directly comparable with the results reported by the challenge organizers and other participating teams (Escalera et al., 2013b; Wu et al., 2013; Bayer and Thierry, 2013).

5.1.4 Structure of Experiments

For the evaluation of the proposed approach we examine the following experimental aspects:

1. First, we present results on the performance of the single modality results; for these the only parameter that we switch on/off is the activity detection, which can be applied on each separate modality; see Section 5.2 and Table 1.

2. Second, we examine the performance in the multimodal cases. This main axis of experiments has as its main reference Table 2 and concerns several aspects, as follows:

   (a) Focus on the basic components of the proposed approach.

8. Note that the Levenshtein distance takes values in $[0, 1]$ and is equivalent to the word error rate.

AD	Single Modalities		
	Aud.	*Skel.*	*HS*
✗	78.4	47.6	13.3
✓	87.2	49.1	20.2

Table 1: Single modalities recognition accuracy %, including Audio (Aud.), Skeleton (Skel.), and Handshape (HS). AD refers to activity detection.

(b) Focus on two stream modality combinations; this serves for both the analysis of our approach, but also provides a more focused comparison with other methods that employ the specific pairs of modalities.

(c) Finally, we provide several fusion based variation experiments, as competitive approaches.

3. Third, we show an indicative example from the actual data, together with its decoding results after applying the proposed approach, compared to the application of a couple of subcomponents.

4. Fourth, we specifically focus on comparisons within the gesture challenge competition. From the list of 17 teams/methods that submitted their results (54 teams participated in total) we review the top-ranked ones, and list their results for comparison. Moreover, we describe the components that each of the top-ranked participants employ, providing also focused comparisons to both our complete approach, and specific cases that match the employed modalities of the other methods. Some cases of our competitive variations can be seen as resembling cases of the other teams' approaches.

5.2 Recognition Results: Single Modalities

In Table 1 we show the recognition results for each independent modality with and without the employment of activity detection (AD). Note that AD is employed for model training, as described in Sections 3.1, 3.3, for each modality. In both cases the audio appears to be the dominant modality in terms of recognition performance. For all modalities, the model-based integration of the activity detectors during training appears to be crucial: they lead to refined temporal boundaries that better align to the actual single-stream activity. In this way we compensate for the fact that the data set annotations may contain non-activity at the start/end of a gesture. By tightening these boundaries we achieve to model in more detail gesture articulation leading to more robustly trained HMMs. This is also projected on the recognition experiments: In all modalities the recognition performance increases, by 8.8%, 1.5% and 6.9% in absolute for the audio, the skeleton and the handshape streams respectively.

	Method/ Exp. Code	Modality	Segm. Method	Classifier/ Modeling	Fusion	Acc. (%)	LD
Others	O1: 1st Rank*	SK, AU	AU:time-domain	HMM, DTW	Late:w-sum	87.24	0.1280
	O2: 2nd Rank†	SK, AU	AU:energy	RF, KNN	Late:posteriors	84.61	0.1540
	O3: 3rd Rank‡	SK, AU	AU:detection	RF, Boosting	Late:w-average	82.90	0.1710
2 Streams	s2-A1	SK,AU	HMM	AD, HMM	Late:SPF	87.9	0.1210
	s2-B1	SK,AU	-	AD,HMM,GRAM	Late:MHS	92.8	0.0720
	s2-A2	HS,AU	HMM	AD, HMM	Late:SPF	87.7	0.1230
	s2-B2	HS,AU	-	AD,HMM,GRAM	Late:MHS	87.5	0.1250
3 Streams	C1	SK,AU,HS	HMM	AD, HMM	Late:SPF	88.5	0.1150
	D1	SK,AU,HS	-	HMM	Late:MHS	85.80	0.1420
	D2	SK,AU,HS	-	AD,HMM	Late:MHS	91.92	0.0808
	D3	SK,AU,HS	-	AD,HMM,GRAM	Late:MHS	93.06	0.0694
	E1	SK,AU,HS	HMM	HMM	Late:MHS+SPF	87.10	0.1290
	E2	SK,AU,HS	HMM	AD,HMM	Late:MHS+SPF	92.28	0.0772
	E3	SK,AU,HS	HMM	AD,HMM,GRAM	Late:MHS+SPF	93.33	0.0670

*(Wu et al., 2013); † (Escalera et al., 2013b); ‡ (Bayer and Thierry, 2013).

Table 2: Comparisons to first-ranked teams in the multimodal challenge recognition ChaLearn 2013, and to several variations of our approach.

5.3 Recognition Results: Multimodal Fusion

For the evaluation of the proposed fusion scheme we focus on several of its basic components. For these we refer to the experiments with codes D1-3,[9] and E1-3 as shown in Table 2. These experiments correspond to the employment of all three modalities, while altering a single component each time, wherever this makes sense.

5.3.1 Main Components and Comparisons

First comes the *MHS component* (see D1-3), which rescores the multimodal hypotheses list employing all three information streams and linearly combining their scores. Comparing with Table 1 the MHS component results in improved performance compared to the monomodal cases, by leading to 38% relative Levenshtein distance reduction (LDR)[10] on average. This improvement is statistically significant, when employing the McNemar's test (Gillick and Cox, 1989), with $p < 0.001$.[11]

Further, the employment of the activity detectors for each modality during training also affects the recognition performance after employing the MHS component, leading to a relative LDR of 38% which is statistically significant ($p < 0.001$); compare D1-D2, E1-E2.

For the N-best multimodal hypothesis rescoring we can either enforce each modality to rescore the exact hypothesis (forced alignment), or allow certain degrees of freedom by

9. The D1-3 notation refers to the D1, D2 and D3 cases.
10. All relative percentages, unless stated otherwise, refer to relative LD reduction (LDR). LDR is equivalent to the known relative word error rate reduction.
11. Statistical significance tests are computed on the raw recognition values and not on the relative improvement scores.

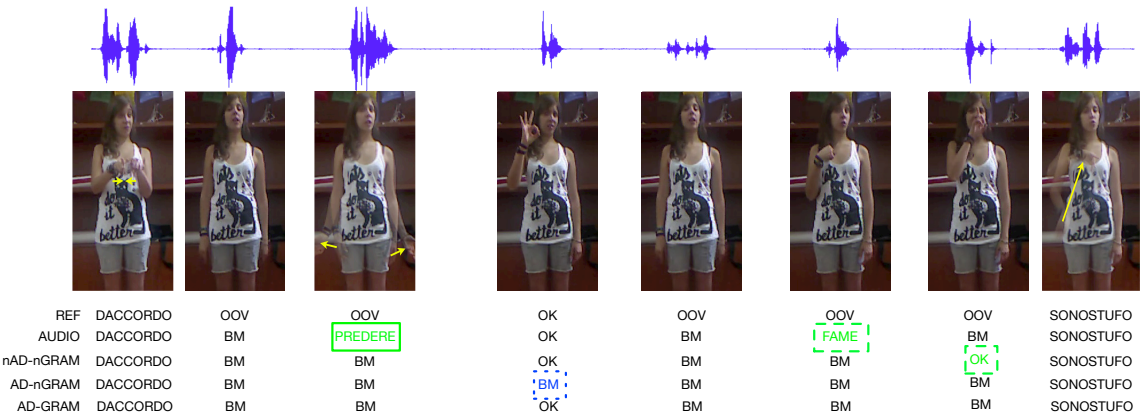


Figure 7: A gesture sequence decoding example. The audio signal is plotted in the top row the and visual modalities (second row) are illustrated via a sequence of images for a gesture sequence. Ground truth transcriptions are denoted by "REF". Decoding results are given for the single-audio modality (AUDIO) and the proposed fusion scheme employing or not the activity detection (AD) or the grammar (GRAM). In nAD-nGRAM we do not employ neither AD nor GRAM during rescoring, in AD-nGRAM we only employ AD but not GRAM and in AD-GRAM both AD and GRAM are employed. Errors are highlighted as: deletions, in blue color, and insertions in green. A background model (bm) models the out-of-vocabulary (OOV) gestures.

employing a *specific grammar* (GRAM) which allows insertions or deletions of either *bm* or *sil* models: By use of the aforementioned grammar *during rescoring* (see D2-D3, E2-E3) we get an additional 14% of relative Levenshtein distance reduction, which is statistically significant ($p < 0.001$). This is due to the fact that the specific grammar accounts for activity or non-activity that does not necessarily occur simultaneously across all different modalities.

In addition, by employing the *SPF component* (E1-3) we further refine the gesture sequence hypothesis by fusing the single-stream models at the segmental level. By comparing corresponding pairs: D1-E1, D2-E2 and D3-E3, we observe that the application of the SPF component increases the recognition performance only slightly; this increase was not found to be statistically significant. The best recognition performance, that is, 93.33%, is obtained after employing the SPF component on top of MHS, together with AD and GRAM (see E3).

On the side, we additionally provide results that account for pairs of modalities; see s2-B1 (AU+SK) and s2-B2 (AU+HS), and for the case of the *MHS component*. These two stream pair results, are comparable with the corresponding 3-stream case of D1 (plus D2-3 for additional components). The rest of the results and pairs are discussed in Section 5.4, where comparisons with other approaches are presented.

5.3.2 Example from the Results

A decoding example is shown in Figure 7. Herein we illustrate both audio and visual modalities for a word sequence accompanied with the ground truth gesture-level transcriptions (row:"REF"). In addition we show the decoding output employing the single-audio modality (AUDIO) and the proposed fusion scheme employing or not two of its basic components: activity detection (AD) and the above mentioned grammar (GRAM). In the row denoted by nAD-nGRAM we do not employ either AD or GRAM during rescoring, in the row AD-nGRAM we only employ AD but not GRAM and in AD-GRAM both AD and grammar are used. As we observe there are several cases where the subject articulates an out-of-vocabulary (OOV) gesture. This indicates the difficulty of the task as these cases should be ignored. By focusing on the recognized word sequence that employs the single-audio modality we notice two insertions ('PREDERE' and 'FAME'). When employing either the nAD-nGRAM or AD-nGRAM the above word insertions are corrected as the visual modality is integrated and helps identifying that these segments correspond to OOV gestures. Finally, both nAD-nGRAM and AD-nGRAM lead to errors which our final proposed approach manages to deal with: nAD-nGRAM causes insertion of "OK", AD-nGRAM of a word deletion "BM". On the contrary, the proposed approach recognizes the whole sentence correctly.

5.4 Comparisons

Next, we first briefly describe the main components of the top-ranked approaches in ChaLearn. This description aims at allowing for focused and fair comparisons between 1) the first-ranked approaches, and 2) variations of our approach.

5.4.1 ChaLearn First-Ranked Approaches

The first-ranked team (IV AMM) (Wu et al., 2013; Escalera et al., 2013b) uses a feature vector based on audio and skeletal information. A simple time-domain end-point detection algorithm based on joint coordinates is applied to segment continuous data sequences into candidate gesture intervals. A HMM is trained with 39-dimension MFCC features and generates confidence scores for each gesture category. A Dynamic Time Warping based skeletal feature classifier is applied to provide complementary information. The confidence scores generated by the two classifiers are firstly normalized and then combined to produce a weighted sum for late fusion. A single threshold approach is employed to classify meaningful gesture intervals from meaningless intervals caused by false detection of speech intervals.

The second-ranked team (WWEIGHT) (Escalera et al., 2013b) combines audio and skeletal information, using both joint spatial distribution and joint orientation. They first search for regions of time with high audio-energy to define time windows that potentially contained a gesture. Feature vectors are defined using a log-spaced audio spectrogram and the joint positions and orientations above the hips. At each time sample the method subtracts the average 3D position of the left and right shoulders from each 3D joint position. Data is down-sampled onto a 5Hz grid. There were 1593 features total (9 time samples x 177 features per time sample). Since some of the detected windows contain distracter gestures, an extra 21st label is introduced, defining the "not in the dictionary" gesture category. For the training of the models they employed an ensemble of randomized decision trees, referred

Rank	Approach	Lev. Dist.	Acc.%	LDR
-	Our	0.0667	93.33	-
1	iva.mm (Wu et al., 2013)	0.12756	87.244	+47.6
2	wweight	0.15387	84.613	+56.6
3	E.T. (Bayer and Thierry, 2013)	0.17105	82.895	+60.9
4	MmM	0.17215	82.785	+61.2
5	pptk	0.17325	82.675	+61.4

Table 3: Our approach in comparison with the first 5 places of the Challenge. We include recognition accuracy (Acc.) %, Levenshtein distance (Lev. Dist., see also text) and relative Levenshtein distance reduction (LDR) (equivalent to the known relative error reduction) compared to the proposed approach (Our).

to as random forests (RF), (Escalera et al., 2013b), and a k-nearest neighbor (KNN) model. The posteriors from these models are averaged with equal weight. Finally, a heuristic is used (12 gestures maximum, no repeats) to convert posteriors to a prediction for the sequence of gestures.

The third-ranked team (ET) (Bayer and Thierry, 2013; Escalera et al., 2013b) combine the output decisions of two approaches. The features considered are based on the skeleton information and the audio signal. First, they look for gesture intervals (unsupervised) using the audio and extract features from these intervals (MFCC). Using these features, they train a random forest (RF) and a gradient boosting classifier. The second approach uses simple statistics (median, var, min, max) on the first 40 frames for each gesture to build the training samples. The prediction phase uses a sliding window. The authors late fuse the two models by creating a weighted average of the outputs.

5.4.2 Comparisons With Other Approaches and Variations

Herein we compare the recognition results of our proposed multimodal recognition and multiple hypotheses fusion framework with other approaches (Escalera et al., 2013b) which have been evaluated in the exact recognition task.[12]

First, let us briefly present an overview of the results (Table 3): Among the numerous groups and approaches that participated we list the first four ones as well as the one we submitted during the challenge, that is "pptk". As shown in Table 3 the proposed approach leads to superior performance with relative LD reduction of at least 47.6%. We note that our updated approach compared to the one submitted during the challenge leads to an improvement of 61.4%, measured in terms of relative LD reduction (LDR). Compared to the approach we submitted during the challenge, the currently proposed scheme: 1) employs activity detection to train single-stream models, 2) applies the SPF on top of the MHS step, 3) introduces the grammar-constrained decoding during hypothesis rescoring and further

12. In all results presented we follow the same blind testing rules that hold in the challenge, in which we have participated (pptk team). In Table 3 we include for common reference the Levenshtein distance (LD) which was also used in the challenge results (Escalera et al., 2013b).

4) incorporates both validation and training data for the final estimation of the model parameters.

Now let us zoom into the details of the comparisons by viewing once again Table 2. In the first three rows, with side label "Others" (O1-3), we summarize the main components of each of the top-ranked approaches. These employ only the two modalities (SK+AU). The experiments with pairs of modalities s2-A1, s2-B1 can be directly compared with O1-3, since they all take advantage of the SK+AU modalities. Their differential concerns 1) the segmentation component, which is explicit for the O1-3; note that the segmentation of s2-A1 is implicit, as a by-product of the HMM recognition. 2) The modeling and recognition/classification component. 3) The fusion component. At the same time, s2-A1/s2-B1 refer to the employment of the proposed components, that is, either SPF or MHS. Specifically, s2-A1 and s2-B1 leads to at least 5% and 43.5% relative LD reduction respectively. Of course our complete system (see rest of variations) leads to even higher improvements.

Other comparisons to our proposed approach and variations are provided after comparing with the SPF-only case, by taking out the contribution of the rescoring component. In the case of all modalities, 3 stream case, (see C1) this is compared to the corresponding matching experiment E2; this (E2) only adds the MHS resulting to an improvement of 32.9% LDR. The GRAM component offers an improvement of 42% LDR (C1 vs. E3). Reduced versions compared to C1, with two-stream combinations can be found by comparing C1 with s2-A1 or s2-A2.

6. Conclusions

We have presented a complete framework for multimodal gesture recognition based on multiple hypotheses fusion, with application in automatic recognition of multimodal gestures. In this we exploit multiple cues in the visual and audio modalities, namely movement, hands' shape and speech. After employing state-of-the-art feature representations, each modality is treated under a common statistical HMM framework: this includes model-based multimodal activity detection, HMM training of gesture-words, and information fusion. Fusion is performed by generating multiple unimodal hypotheses, which after constrained rescoring and weighted combination result in the multimodally best hypothesis. Then, segmental parallel fusion across all modalities refines the final result. On the way, we employ gesture/speech background (bm) and silence (sil) models, which are initialized during the activity detection stage. This procedure allows us to train our HMMs more accurately by getting tighter temporal segmentation boundaries.

The recognition task we dealt with contains parallel gestures and spoken words, articulated freely, containing multiple sources of multimodal variability, and with on purpose false alarms. The overall framework is evaluated in a demanding multimodal data set (Escalera et al., 2013b) achieving 93.3% word accuracy. The results are compared with several approaches that participated in the related challenge (Escalera et al., 2013a), under the same blind testing conditions, leading to at least 47.6% relative Levenshtein distance reduction (equivalent to relative word error rate reduction) compared to the first-ranked team (Wu et al., 2013).

The power of the proposed fusion scheme stems from both its uniform across modalities probabilistic nature and its late character together with the multiple passes of monomodal

decoding, fusion of the hypotheses, and then parallel fusion. Apart from the experimental evidence, these features render it appealing for extensions and exploitation in multiple directions: First, the method itself can be advanced by generalizing the approach towards an iterative fusion scheme, that gives feedback back to the training/refinement stage of the statistical models. Moreover in the current generative framework, we ignore statistical dependencies across cues/modalities. These could further be examined. Second, it can be advanced by incorporating in the computational modeling specific gesture theories, e.g., from linguistics, for the gesture per se or in its multimodal version; taxonomies of gestures, e.g., that describe deictic, motor, iconic and metaphoric cases. Such varieties of cases can be systematically studied with respect to their role. This could be achieved via automatic processing of multitudes of existing data sets, which elaborate more complex speech-gesture issues, leading to valuable analysis results. Then, apart from the linguistic role of gesture, its relation to other aspects, such as, psychological, behavioral socio-cultural, or communicative, to name but a few, could further be exploited. To conclude, given the potential of the proposed approach, the acute interdisciplinary interest in multimodal gesture calls for further exploration and advancements.

Acknowledgements

This research work was supported by the project "COGNIMUSE" which is implemented under the "ARISTEIA" Action of the Operational Program Education and Lifelong Learning and is co-funded by the European Social Fund and Greek National Resources. It was also partially supported by the European Union under the project "MOBOT" with grant FP7-ICT-2011-9 2.1 - 600796. The authors want to gratefully thank Georgios Pavlakos for his contribution in previous, earlier stages, of this work.

References

U. Agris, J. Zieren, U. Canzler, B. Bauer, and K.-F. Kraiss. Recent developments in visual sign language recognition. *Universal Access in the Information Society*, 6:323–362, 2008.

J. Alon, V. Athitsos, Q. Yuan, and S. Sclaroff. A unified framework for gesture recognition and spatiotemporal gesture segmentation. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 31(9):1685–1699, 2009.

A. Argyros and M. Lourakis. Real time tracking of multiple skin-colored objects with a possibly moving camera. In *Proc. Europ. Conf. on Computer Vision*, 2004.

B. Bauer and K. F. Kraiss. Towards an automatic sign language recognition system using subunits. In *Proc. of Int'l Gest. Wrksp*, volume 2298, pages 64–75, 2001.

I. Bayer and S. Thierry. A multi modal approach to gesture recognition from audio and video data. In *Proc. of the 15th ACM Int'l Conf. on Multimodal Interaction*, pages 461–466. ACM, 2013.

P. Bernardis and M. Gentilucci. Speech and gesture share the same communication system. *Neuropsychologia*, 44(2):178–190, 2006.

N. D. Binh, E. Shuichi, and T. Ejima. Real-time hand tracking and gesture recognition system. In *Proc. of Int'l Conf. on Graphics, Vision and Image Processing (GVIP)*, pages 19–21, 2005.

A. F. Bobick and J. W. Davis. The recognition of human movement using temporal templates. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 23(3):257–267, 2001.

R. A. Bolt. "Put-that-there": Voice and gesture at the graphics interface. In *Proc. of the 7th Annual Conf. on Computer Graphics and Interactive Techniques*, volume 14. ACM, 1980.

H. Bourlard and S. Dupont. Subband-based speech recognition. In *Proc. Int'l Conf. on Acoustics, Speech and Sig. Proc.*, volume 2, pages 1251–1254. IEEE, 1997.

K. Bousmalis, L. Morency, and M. Pantic. Modeling hidden dynamics of multimodal cues for spontaneous agreement and disagreement recognition. In *Proc. Int'l Conf. on Autom. Face & Gest. Rec.*, pages 746–752. IEEE, 2011.

P. Buehler, M. Everingham, and A. Zisserman. Learning sign language by watching TV (using weakly aligned subtitles). In *Proc. Int'l Conf. on Comp. Vis. & Patt. Rec.*, 2009.

S. Celebi, A. S. Aydin, T. T. Temiz, and T. Arici. Gesture recognition using skeleton data with weighted dynamic time warping. *Computer Vision Theory and Applications*, 2013.

F.-S. Chen, C.-M. Fu, and C.-L. Huang. Hand gesture recognition using a real-time tracking method and hidden Markov models. *Image and Vis. Computing*, 21(8):745–758, 2003.

X. Chen and M. Koskela. Online rgb-d gesture recognition with extreme learning machines. In *Proc. of the 15th ACM Int'l Conf. on Multimodal Interaction*, pages 467–474. ACM, 2013.

Y. L. Chow and R. Schwartz. The n-best algorithm: An efficient procedure for finding top n sentence hypotheses. In *Proc. of the Workshop on Speech and Natural Language*, pages 199–202. Association for Computational Linguistics, 1989.

S. Conseil, S. Bourennane, and L. Martin. Comparison of Fourier descriptors and Hu moments for hand posture recognition. In *Proc. European Conf. on Signal Processing*, 2007.

Y. Cui and J. Weng. Appearance-based hand sign recognition from intensity image sequences. *Comp. Vis. and Im. Undrst.*, 78(2):157–176, 2000.

N. Dalal and B. Triggs. Histogram of oriented gradients for human detection. In *Proc. Int'l Conf. on Comp. Vis. & Patt. Rec.*, 2005.

W. Du and J. Piater. Hand modeling and tracking for video-based sign language recognition by robust principal component analysis. In *Proc. ECCV Wksp on Sign, Gest. and Activity*, September 2010.

S. Escalera, J. Gonzàlez, X. Baró, M. Reyes, I. Guyon, V. Athitsos, H. Escalante, L. Sigal, A. Argyros, C. Sminchisescu, R. Bowden, and S. Sclaroff. ChaLearn multi-modal gesture recognition 2013: grand challenge and workshop summary. In *Proc. of the 15th ACM on Int'l Conf. on Multimodal Interaction*, pages 365–368. ACM, 2013a.

S. Escalera, J. Gonzlez, X. Bar, M. Reyes, O. Lopes, I. Guyon, V. Athitsos, and H.J. Escalante. Multi-modal Gesture Recognition Challenge 2013: Dataset and Results. In *15th ACM Int'l Conf. on Multimodal Interaction (ICMI), ChaLearn Challenge and Wrksp on Multi-modal Gesture Recognition*. ACM, 2013b.

J. Foote. An overview of audio information retrieval. *Multimedia Systems*, 7(1):2–10, 1999. URL `http://link.springer.com/article/10.1007/s005300050106`.

L. Gillick and S.J. Cox. Some statistical issues in the comparison of speech recognition algorithms. In *Proc. Int'l Conf. on Acoustics, Speech and Sig. Proc.*, pages 532–535 vol.1, may 1989.

H. Glotin, D. Vergyr, C. Neti, G. Potamianos, and J. Luettin. Weighting schemes for audio-visual fusion in speech recognition. In *Proc. Int'l Conf. on Acoustics, Speech and Sig. Proc.*, volume 1, pages 173–176. IEEE, 2001.

B. Habets, S. Kita, Z. Shao, A. Özyurek, and P. Hagoort. The role of synchrony and ambiguity in speech–gesture integration during comprehension. *Journal of Cognitive Neuroscience*, 23(8):1845–1854, 2011.

J. Han, G. Awad, and A. Sutherland. Modelling and segmenting subunits for sign language recognition based on hand motion analysis. *Patt. Rec. Letters*, 30:623–633, 2009.

H. Hermansky. Perceptual linear predictive (plp) analysis of speech. *Journal of the Acoustical Society of America*, 87(4):1738–1752, 1990.

A. Hernández-Vela, M. Á. Bautista, X. Perez-Sala, V. Ponce-López, S. Escalera, X. Baró, O. Pujol, and C. Angulo. Probability-based dynamic time warping and bag-of-visual-and-depth-words for human gesture recognition in rgb-d. *Patt. Rec. Letters*, 2013.

C.-L. Huang and S.-H. Jeng. A model-based hand gesture recognition system. *Machine Vision and Application*, 12(5):243–258, 2001.

M. Isard and A. Blake. Condensation-conditional density propagation for visual tracking. *Int'l Journal of Computer Vision*, 29(1):5–28, 1998.

M. Iverson, J. and S. Goldin-Meadow. Why people gesture when they speak. *Nature*, 396 (6708):228–228, 1998.

A. Jaimes and N. Sebe. Multimodal human–computer interaction: A survey. *Comp. Vis. and Im. Undrst.*, 108(1):116–134, 2007.

S. D. Kelly, A. Özyürek, and E. Maris. Two sides of the same coin speech and gesture mutually interact to enhance comprehension. *Psychological Science*, 21(2):260–267, 2010.

A. Kendon. *Gesture: Visible Action as Utterance*. Cambridge University Press, 2004.

W. Kong and S. Ranganath. Sign language phoneme transcription with rule-based hand trajectory segmentation. *J. Signal Proc. Sys.*, 59:211–222, 2010.

I. Laptev, M. Marszalek, and B. Schmid, C.and Rozenfeld. Learning realistic human actions from movies. In *Proc. Int'l Conf. on Comp. Vis. & Patt. Rec.*, pages 1–8. IEEE, 2008.

H-K. Lee and J-H. Kim. An HMM-based threshold model approach for gesture recognition. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 21(10):961–973, 1999.

J. Li and N. M. Allinson. Simultaneous gesture segmentation and recognition based on forward spotting accumulative hmms. *Pattern Recognition*, 40(11):3012–3026, 2007.

J. Li and N. M. Allinson. A comprehensive review of current local features for computer vision. *Neurocomputing*, 71(10):1771–1787, 2008.

D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. Int'l Conf. on Comp. Vis.*, pages 1150–1157, 1999.

P. Maragos, P. Gros, A. Katsamanis, and Papandreou G. Cross-modal integration for performance improving in multimedia: A review. In P. Maragos, A. Potamianos, and P. Gros, editors, *Multimodal Processing and Interaction: Audio, Video, Text*, chapter 1, pages 3–48. Springer-Verlag, New York, 2008.

D. McNeill. *Hand and Mind: What Gestures Reveal About Thought*. University of Chicago Press, 1992.

M. Miki, N. Kitaoka, C. Miyajima, T. Nishino, and K. Takeda. Improvement of multimodal gesture and speech recognition performance using time intervals between gestures and accompanying speech. *EURASIP Journal on Audio, Speech, and Music Processing*, 2014 (1):17, 2014. URL `http://link.springer.com/article/10.1186/1687-4722-2014-2`.

D. Morris, P. Collett, P. Marsh, and O'Shaughnessy M. *Gestures: Their Origins and Distribution*. Stein and Day, 1979.

Y. Nam and K. Wohn. Recognition of space-time hand-gestures using hidden Markov model. In *ACM Symposium on Virtual Reality Software and Technology*, pages 51–58, 1996.

K. Nandakumar, K. W. Wan, S. Chan, W. Ng, J. G. Wang, and W. Y. Yau. A multi-modal gesture recognition system using audio, video, and skeletal joint data. In *Proc. of the 15th ACM Int'l Conf. on Multimodal Interaction*, pages 475–482. ACM, 2013.

N. Neverova, C. Wolf, G. Paci, G. Sommavilla, G. Taylor, and F. Nebout. A multi-scale approach to gesture detection and recognition. In *Proc. of the IEEE Int'l Conf. on Computer Vision Wrksp*, pages 484–491, 2013.

E.-J. Ong and R. Bowden. A boosted classifier tree for hand shape detection. In *Proc. Int'l Conf. on Autom. Face & Gest. Rec.*, pages 889–894. IEEE, 2004.

M. Ostendorf, A. Kannan, S. Austin, O. Kimball, R. M. Schwartz, and J. R. Rohlicek. Integration of Diverse Recognition Methodologies Through Reevaluation of N-Best Sentence Hypotheses. In *HLT*, 1991.

S. Oviatt and P. Cohen. Perceptual user interfaces: multimodal interfaces that process what comes naturally. *Communications of the ACM*, 43(3):45–53, 2000.

G. Papandreou, A. Katsamanis, V. Pitsikalis, and P. Maragos. Adaptive multimodal fusion by uncertainty compensation with application to audiovisual speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(3):423–435, 2009.

V. Pitsikalis, S. Theodorakis, C. Vogler, and P. Maragos. Advances in phonetics-based sub-unit modeling for transcription alignment and sign language recognition. In *IEEE CVPR Wksp on Gest. Rec.*, 2011.

I. Poddar, Y. Sethi, E. Ozyildiz, and R. Sharma. Toward natural gesture/speech HCI: A case study of weather narration. In *Proc. Wrksp on Perceptual User Interfaces*, 1998.

V. Ponce-López, S. Escalera, and X. Baró. Multi-modal social signal analysis for predicting agreement in conversation settings. In *Proc. of the 15th ACM Int'l Conf. on Multimodal Interaction*, pages 495–502. ACM, 2013.

G. Potamianos, C. Neti, J. Luettin, and I. Matthews. Audio-visual automatic speech recognition: An overview. *Issues in Visual and Audio-Visual Speech Processing*, 22:23, 2004.

L.R. Rabiner and B.H. Juang. *Fundamentals of Speech Recognition*. Prentice Hall, 1993.

Z. Ren, J. Yuan, and Z. Zhang. Robust hand gesture recognition based on finger-earth mover's distance with a commodity depth camera. In *Proc. of the 19th ACM Int'l Conf. on Multimedia*, pages 1093–1096. ACM, 2011.

R. C. Rose. Discriminant wordspotting techniques for rejecting non-vocabulary utterances in unconstrained speech. In *Proc. Int'l Conf. on Acoustics, Speech and Sig. Proc.*, volume 2, pages 105–108. IEEE, 1992. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=226109`.

R. C. Rose and D. B. Paul. A hidden Markov model based keyword recognition system. In *Proc. Int'l Conf. on Acoustics, Speech and Sig. Proc.*, pages 129–132, 1990. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=115555`.

A. Roussos, S. Theodorakis, V. Pitsikalis, and P. Maragos. Dynamic affine-invariant shape-appearance handshape features and classification in sign language videos. *Journal of Machine Learning Research*, 14(1):1627–1663, 2013.

S. Ruffieux, D. Lalanne, and E. Mugellini. ChAirGest: A Challenge for Multimodal Mid-air Gesture Recognition for Close HCI. In *Proc. of the 15th ACM Int'l Conf. on Multimodal Interaction*, ICMI '13, pages 483–488, New York, NY, USA, 2013. ACM.

S. Ruffieux, D. Lalanne, E. Mugellini, and O. A. Khaled. A Survey of Datasets for Human Gesture Recognition. In *Human-Computer Interaction. Advanced Interaction Modalities and Techniques*, pages 337–348. Springer, 2014.

R. Sharma, M. Yeasin, N. Krahnstoever, I. Rauschert, G. Cai, I. Brewer, A. M MacEachren, and K. Sengupta. Speech-gesture driven multimodal interfaces for crisis management. *Proc. of the IEEE*, 91(9):1327–1354, 2003.

S. Shimojo and L. Shams. Sensory modalities are not separate modalities: plasticity and interactions. *Current Opinion in Neurobiology*, 11(4):505–509, 2001.

J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124, 2013.

R. Shwartz and S. Austin. A comparison of several approximate algorithms for finding multiple N-Best sentence hypotheses. In *Proc. Int'l Conf. on Acoustics, Speech and Sig. Proc.*, 1991.

Y. C. Song, H. Kautz, J. Allen, M. Swift, Y. Li, J. Luo, and C. Zhang. A Markov logic framework for recognizing complex events from multimodal data. In *Proc. of the 15th ACM Int'l Conf. on Multimodal Interaction*, pages 141–148. ACM, 2013.

T. Starner, J. Weaver, and A. Pentland. Real-time american sign language recognition using desk and wearable computer based video. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 20(12):1371–1375, Dec. 1998.

L. N. Tan, B. J. Borgstrom, and A. Alwan. Voice activity detection using harmonic frequency components in likelihood ratio test. In *Proc. Int'l Conf. on Acoustics, Speech and Sig. Proc.*, pages 4466–4469. IEEE, 2010.

N. Tanibata, N. Shimada, and Y. Shirai. Extraction of hand features for recognition of sign language words. In *Proc. Int'l Conf. on Vision Interface*, pages 391–398, 2002.

S. Theodorakis, V. Pitsikalis, and P. Maragos. Dynamic-Static Unsupervised Sequentiality, Statistical Subunits and Lexicon for Sign Language Recognition. *Imave and Vision Computing*, 32(8):533549, 2014.

M. Turk. Multimodal interaction: A review. *Patt. Rec. Letters*, 36:189–195, 2014.

C. Vogler and D. Metaxas. A framework for recognizing the simultaneous aspects of american sign language. *Comp. Vis. and Im. Undrst.*, 81:358, 2001.

S. B. Wang, A. Quattoni, L. Morency, D. Demirdjian, and T. Darrell. Hidden conditional random fields for gesture recognition. In *Proc. Int'l Conf. on Comp. Vis. & Patt. Rec.*, volume 2, pages 1521–1527. IEEE, 2006.

D. Weimer and S. Ganapathy. A synthetic visual environment with hand gesturing and voice input. In *ACM SIGCHI Bulletin*, volume 20, pages 235–240. ACM, 1989.

L. D Wilcox and M. Bush. Training and search algorithms for an interactive wordspotting system. In *Proc. Int'l Conf. on Acoustics, Speech and Sig. Proc.*, volume 2, pages 97–100. IEEE, 1992.

J. Wilpon, L. R. Rabiner, C.-H. Lee, and E. R. Goldman. Automatic recognition of keywords in unconstrained speech using hidden Markov models. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 38(11):1870–1878, 1990.

A. Wilson and A. Bobick. Parametric hidden Markov models for gesture recognition. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 21:884–900, 1999.

J. Wu, J. Cheng, C. Zhao, and H. Lu. Fusing multi-modal features for gesture recognition. In *Proc. of the 15th ACM Int'l Conf. on Multimodal Interaction*, pages 453–460. ACM, 2013.

M.-H. Yang, N. Ahuja, and M. Tabb. Extraction of 2d motion trajectories and its application to hand gesture recognition. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 24(8):1061–1074, Aug. 2002.

S. Young, G. Evermann, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland. *The HTK Book*. Entropic Cambridge Research Laboratory, Cambridge, United Kingdom, 2002.

Journal of Machine Learning Research 16 (2015) 285-322 Submitted 6/14; Published 2/15

An Asynchronous Parallel Stochastic Coordinate Descent Algorithm

Ji Liu JI.LIU.UWISC@GMAIL.COM
Stephen J. Wright SWRIGHT@CS.WISC.EDU
Department of Computer Sciences
University of Wisconsin-Madison
Madison, WI 53706-1685

Christopher Ré CHRISMRE@CS.STANFORD.EDU
Department of Computer Science
Stanford University
353 Serra Mall
Stanford, CA 94305-9025

Victor Bittorf BITTORF@CS.WISC.EDU
Srikrishna Sridhar SRIKRIS@CS.WISC.EDU
Department of Computer Sciences
University of Wisconsin-Madison
Madison, WI 53706-1685

Editor: Leon Bottou

Abstract

We describe an asynchronous parallel stochastic coordinate descent algorithm for minimizing smooth unconstrained or separably constrained functions. The method achieves a linear convergence rate on functions that satisfy an essential strong convexity property and a sublinear rate ($1/K$) on general convex functions. Near-linear speedup on a multicore system can be expected if the number of processors is $O(n^{1/2})$ in unconstrained optimization and $O(n^{1/4})$ in the separable-constrained case, where n is the number of variables. We describe results from implementation on 40-core processors.

Keywords: asynchronous parallel optimization, stochastic coordinate descent

1. Introduction

Consider the convex optimization problem

$$\min_{x\in\Omega} \quad f(x), \tag{1}$$

where $\Omega \subset \mathbb{R}^n$ is a closed convex set and f is a smooth convex mapping from an open neighborhood of Ω to $\mathbb{R}$. We consider two particular cases of Ω in this paper: the unconstrained case $\Omega = \mathbb{R}^n$, and the separable case

$$\Omega = \Omega_1 \times \Omega_2 \times \ldots \times \Omega_n, \tag{2}$$

where each Ω_i, $i = 1, 2, \ldots, n$ is a closed subinterval of the real line.

Formulations of the type (1,2) arise in many data analysis and machine learning problems, for example, support vector machines (linear or nonlinear dual formulation) (Cortes and Vapnik, 1995), LASSO (after decomposing x into positive and negative parts) (Tibshirani, 1996), and logistic regression. Algorithms based on gradient and approximate or partial gradient information have proved effective in these settings. We mention in particular gradient projection and its accelerated variants (Nesterov, 2004), accelerated proximal gradient methods for regularized objectives (Beck and Teboulle, 2009), and stochastic gradient methods (Nemirovski et al., 2009; Shamir and Zhang, 2013). These methods are inherently serial, in that each iteration depends on the result of the previous iteration. Recently, parallel multicore versions of stochastic gradient and stochastic coordinate descent have been described for problems involving large data sets; see for example Niu et al. (2011); Richtárik and Takáč (2012b); Avron et al. (2014).

This paper proposes an asynchronous stochastic coordinate descent (AsySCD) algorithm for convex optimization. Each step of AsySCD chooses an index $i \in \{1, 2, \ldots, n\}$ and subtracts a short, constant, positive multiple of the ith partial gradient $\nabla_i f(x) := \partial f/\partial x_i$ from the ith component of x. When separable constraints (2) are present, the update is "clipped" to maintain feasibility with respect to Ω_i. Updates take place in parallel across the cores of a multicore system, without any attempt to synchronize computation between cores. We assume that there is a bound τ on the age of the updates, that is, no more than τ updates to x occur between the time at which a processor reads x (and uses it to evaluate one element of the gradient) and the time at which this processor makes its update to a single element of x. (A similar model of parallel asynchronous computation was used in Hogwild! (Niu et al., 2011).) Our implementation, described in Section 6, is a little more complex than this simple model would suggest, as it is tailored to the architecture of the Intel Xeon machine that we use for experiments.

We show that linear convergence can be attained if an "essential strong convexity" property (3) holds, while sublinear convergence at a "$1/K$" rate can be proved for general convex functions. Our analysis also defines a sufficient condition for near-linear speedup in the number of cores used. This condition relates the value of delay parameter τ (which relates to the number of cores / threads used in the computation) to the problem dimension n. A parameter that quantifies the cross-coordinate interactions in ∇f also appears in this relationship. When the Hessian of f is nearly diagonal, the minimization problem can almost be separated along the coordinate axes, so higher degrees of parallelism are possible.

We review related work in Section 2. Section 3 specifies the proposed algorithm. Convergence results for unconstrained and constrained cases are described in Sections 4 and 5, respectively, with proofs given in the appendix. Computational experience is reported in Section 6. We discuss several variants of AsySCD in Section 7. Some conclusions are given in Section 8.

1.1 Notation and Assumption

We use the following notation.

- $e_i \in \mathbb{R}^n$ denotes the ith natural basis vector $(0, \ldots, 0, 1, 0, \ldots, 0)^T$ with the "1" in the ith position.
- $\|\cdot\|$ denotes the Euclidean norm $\|\cdot\|_2$.

- $S \subset \Omega$ denotes the set on which f attains its optimal value, which is denoted by f^*.
- $\mathcal{P}_S(\cdot)$ and $\mathcal{P}_\Omega(\cdot)$ denote Euclidean projection onto S and Ω, respectively.
- We use x_i for the ith element of x, and $\nabla_i f(x)$ for the ith element of the gradient vector $\nabla f(x)$.
- We define the following *essential strong convexity* condition for a convex function f with respect to the optimal set S, with parameter $l > 0$:

$$f(x) - f(y) \geq \langle \nabla f(y), x - y \rangle + \frac{l}{2}\|x - y\|^2 \quad \text{for all } x, y \in \Omega \text{ with } \mathcal{P}_S(x) = \mathcal{P}_S(y). \tag{3}$$

  This condition is significantly weaker than the usual strong convexity condition, which requires the inequality to hold for *all* $x, y \in \Omega$. In particular, it allows for non-singleton solution sets S, provided that f increases at a uniformly quadratic rate with distance from S. (This property is noted for convex quadratic f in which the Hessian is rank deficient.) Other examples of essentially strongly convex functions that are not strongly convex include:

  - $f(Ax)$ with arbitrary linear transformation A, where $f(\cdot)$ is strongly convex;
  - $f(x) = \max(a^T x - b, 0)^2$, for $a \neq 0$.

- Define L_{res} as the *restricted Lipschitz constant* for ∇f, where the "restriction" is to the coordinate directions: We have

$$\|\nabla f(x) - \nabla f(x + te_i)\| \leq L_{\text{res}}|t|, \quad \text{for all } i = 1, 2, \ldots, n \text{ and } t \in \mathbb{R}, \text{ with } x, x + te_i \in \Omega.$$

- Define L_i as the *coordinate Lipschitz constant* for ∇f in the ith coordinate direction: We have

$$f(x + te_i) - f(x) \leq \langle \nabla_i f(x),\ t \rangle + \frac{L_i}{2}t^2, \quad \text{for } i \in \{1, 2, \ldots, n\}, \text{ and } x, x + te_i \in \Omega,$$

  or equivalently

$$|\nabla_i f(x) - \nabla_i f(x + te_i)| \leq L_i|t|.$$

- $L_{\max} := \max_{i=1,2,\ldots,n} L_i$.

Note that $L_{\text{res}} \geq L_{\max}$.

We use $\{x_j\}_{j=0,1,2,\ldots}$ to denote the sequence of iterates generated by the algorithm from starting point x_0. Throughout the paper, we make the following assumption.

Assumption 1

- *The optimal solution set S of (1) is nonempty.*
- *The radius of the iterate set $\{x_j\}_{j=0,1,2,\ldots}$ defined by*

$$R := \sup_{j=0,1,2,\ldots} \|x_j - \mathcal{P}_S(x_j)\|$$

  is bounded, that is, $R < +\infty$.

1.2 Lipschitz Constants

The nonstandard Lipschitz constants L_{res}, $L_{\max}$, and L_i, $i = 1, 2, \ldots, n$ defined above are crucial in the analysis of our method. Besides bounding the nonlinearity of f along various directions, these quantities capture the interactions between the various components in the gradient ∇f, as quantified in the off-diagonal terms of the Hessian $\nabla^2 f(x)$ — although the stated conditions do not require this matrix to exist.

We have noted already that $L_{\text{res}}/L_{\max} \geq 1$. Let us consider upper bounds on this ratio under certain conditions. When f is twice continuously differentiable, we have

$$L_i = \sup_{x \in \Omega} \max_{i=1,2,\ldots,n} [\nabla^2 f(x)]_{ii}.$$

Since $\nabla^2 f(x) \succeq 0$ for $x \in \Omega$, we have that

$$|[\nabla^2 f(x)]_{ij}| \leq \sqrt{L_i L_j} \leq L_{\max}, \quad \forall\, i, j = 1, 2, \ldots, n.$$

Thus L_{res}, which is a bound on the largest column norm for $\nabla^2 f(x)$ over all $x \in \Omega$, is bounded by $\sqrt{n} L_{\max}$, so that

$$\frac{L_{\text{res}}}{L_{\max}} \leq \sqrt{n}.$$

If the Hessian is structurally sparse, having at most p nonzeros per row/column, the same argument leads to $L_{\text{res}}/L_{\max} \leq \sqrt{p}$.

If $f(x)$ is a convex quadratic with Hessian Q, we have

$$L_{\max} = \max_i Q_{ii}, \quad L_{\text{res}} = \max_i \|Q_{\cdot i}\|_2,$$

where $Q_{\cdot i}$ denotes the ith column of Q. If Q is diagonally dominant, we have for any column i that

$$\|Q_{\cdot i}\|_2 \leq Q_{ii} + \|[Q_{ji}]_{j \neq i}\|_2 \leq Q_{ii} + \sum_{j \neq i} |Q_{ji}| \leq 2 Q_{ii},$$

which, by taking the maximum of both sides, implies that $L_{\text{res}}/L_{\max} \leq 2$ in this case.

Finally, consider the objective $f(x) = \frac{1}{2}\|Ax - b\|^2$ and assume that $A \in \mathbb{R}^{m \times n}$ is a random matrix whose entries are i.i.d from $\mathcal{N}(0, 1)$. The diagonals of the Hessian are $A_{\cdot i}^T A_{\cdot i}$ (where $A_{\cdot i}$ is the ith column of A), which have expected value m, so we can expect $L_{\max}$ to be not less than m. Recalling that L_{res} is the maximum column norm of $A^T A$, we have

$$\begin{aligned}
\mathbb{E}(\|A^T A_{\cdot i}\|) &\leq \mathbb{E}(|A_{\cdot i}^T A_{\cdot i}|) + \mathbb{E}(\|[A_{\cdot j}^T A_{\cdot i}]_{j \neq i}\|) \\
&= m + \mathbb{E} \sqrt{\sum_{j \neq i} |A_{\cdot j}^T A_{\cdot i}|^2} \\
&\leq m + \sqrt{\sum_{j \neq i} \mathbb{E} |A_{\cdot j}^T A_{\cdot i}|^2} \\
&= m + \sqrt{(n-1)m},
\end{aligned}$$

where the second inequality uses Jensen's inequality and the final equality uses

$$\mathbb{E}(|A_{\cdot j}^T A_{\cdot i}|^2) = \mathbb{E}(A_{\cdot j}^T \mathbb{E}(A_{\cdot i} A_{\cdot i}^T) A_{\cdot j}) = \mathbb{E}(A_{\cdot j}^T I A_{\cdot j}) = \mathbb{E}(A_{\cdot j}^T A_{\cdot j}) = m.$$

We can thus estimate the upper bound on $L_{\text{res}}/L_{\max}$ roughly by $1 + \sqrt{n/m}$ for this case.

2. Related Work

This section reviews some related work on coordinate relaxation and stochastic gradient algorithms.

Among *cyclic coordinate descent* algorithms, Tseng (2001) proved the convergence of a block coordinate descent method for nondifferentiable functions with certain conditions. Local and global linear convergence were established under additional assumptions, by Luo and Tseng (1992) and Wang and Lin (2014), respectively. Global linear (sublinear) convergence rate for strongly (weakly) convex optimization was proved by Beck and Tetruashvili (2013). Block-coordinate approaches based on proximal-linear subproblems are described by Tseng and Yun (2009, 2010). Wright (2012) uses acceleration on reduced spaces (corresponding to the optimal manifold) to improve the local convergence properties of this approach.

Stochastic coordinate descent is almost identical to cyclic coordinate descent except selecting coordinates in a random manner. Nesterov (2012) studied the convergence rate for a stochastic block coordinate descent method for unconstrained and separably constrained convex smooth optimization, proving linear convergence for the strongly convex case and a sublinear $1/K$ rate for the convex case. Extensions to minimization of composite functions are described by Richtárik and Takáč (2012a) and Lu and Xiao (2013).

Synchronous parallel methods distribute the workload and data among multiple processors, and coordinate the computation among processors. Ferris and Mangasarian (1994) proposed to distribute variables among multiple processors and optimize concurrently over each subset. The synchronization step searches the affine hull formed by the current iterate and the points found by each processor. Similar ideas appeared in (Mangasarian, 1995), with a different synchronization step. Goldfarb and Ma (2012) considered a multiple splitting algorithm for functions of the form $f(x) = \sum_{k=1}^{N} f_k(x)$ in which N models are optimized separately and concurrently, then combined in an synchronization step. The alternating direction method-of-multiplier (ADMM) framework (Boyd et al., 2011) can also be implemented in parallel. This approach dissects the problem into multiple subproblems (possibly after replication of primal variables) and optimizes concurrently, then synchronizes to update multiplier estimates. Duchi et al. (2012) described a subgradient dual-averaging algorithm for partially separable objectives, with subgradient evaluations distributed between cores and combined in ways that reflect the structure of the objective. Parallel stochastic gradient approaches have received broad attention; see Agarwal and Duchi (2011) for an approach that allows delays between evaluation and update, and Cotter et al. (2011) for a minibatch stochastic gradient approach with Nesterov acceleration. Shalev-Shwartz and Zhang (2013) proposed an accelerated stochastic dual coordinate ascent method.

Among *synchronous parallel methods for (block) coordinate descent*, Richtárik and Takáč (2012b) described a method of this type for convex composite optimization problems. All processors update randomly selected coordinates or blocks, concurrently and synchronously, at each iteration. Speedup depends on the sparsity of the data matrix that defines the loss functions. Several variants that select blocks greedily are considered by Scherrer et al. (2012) and Peng et al. (2013). Yang (2013) studied the parallel stochastic dual coordinate ascent method and emphasized the balance between computation and communication.

We turn now to *asynchronous parallel methods*. Bertsekas and Tsitsiklis (1989) introduced an asynchronous parallel implementation for general fixed point problems $x = q(x)$ over a separable convex closed feasible region. (The optimization problem (1) can be formulated in this way by defining $q(x) := \mathcal{P}_\Omega[(I - \alpha\nabla f)(x)]$ for some fixed $\alpha > 0$.) Their analysis allows inconsistent reads for x, that is, the coordinates of the read x have different "ages." Linear convergence is established if all ages are bounded and $\nabla^2 f(x)$ satisfies a diagonal dominance condition guaranteeing that the iteration $x = q(x)$ is a maximum-norm contraction mapping for sufficient small α. However, this condition is strong — stronger, in fact, than the strong convexity condition. For convex quadratic optimization $f(x) = \frac{1}{2}x^T Ax + bx$, the contraction condition requires diagonal dominance of the Hessian: $A_{ii} > \sum_{i \neq j} |A_{ij}|$ for all $i = 1, 2, \ldots, n$. By comparison, AsySCD guarantees linear convergence rate under the essential strong convexity condition (3), though we do not allow inconsistent read. (We require the vector x used for each evaluation of $\nabla_i f(x)$ to have existed at a certain point in time.)

Hogwild! (Niu et al., 2011) is a lock-free, asynchronous parallel implementation of a stochastic-gradient method, targeted to a multicore computational model similar to the one considered here. Its analysis assumes consistent reading of x, and it is implemented without locking or coordination between processors. Under certain conditions, convergence of Hogwild! approximately matches the sublinear $1/K$ rate of its serial counterpart, which is the constant-steplength stochastic gradient method analyzed in Nemirovski et al. (2009).

We also note recent work by Avron et al. (2014), who proposed an asynchronous linear solver to solve $Ax = b$ where A is a symmetric positive definite matrix, proving a linear convergence rate. Both inconsistent- and consistent-read cases are analyzed in this paper, with the convergence result for inconsistent read being slightly weaker.

3. Algorithm

In AsySCD, multiple processors have access to a shared data structure for the vector x, and each processor is able to compute a randomly chosen element of the gradient vector $\nabla f(x)$. Each processor repeatedly runs the following coordinate descent process (the steplength parameter γ is discussed further in the next section):

R: Choose an index $i \in \{1, 2, \ldots, n\}$ at random, read x, and evaluate $\nabla_i f(x)$;

U: Update component i of the shared x by taking a step of length $\gamma / L_{\max}$ in the direction $-\nabla_i f(x)$.

Since these processors are being run concurrently and without synchronization, x may change between the time at which it is read (in step R) and the time at which it is updated (step U). We capture the system-wide behavior of AsySCD in Algorithm 1. There is a global counter j for the total number of updates; x_j denotes the state of x after j updates. The index $i(j) \in \{1, 2, \ldots, n\}$ denotes the component updated at step j. $k(j)$ denotes the x-iterate at which the update applied at iteration j was calculated. Obviously, we have $k(j) \leq j$, but we assume that the delay between the time of evaluation and updating is bounded uniformly by a positive integer τ, that is, $j - k(j) \leq \tau$ for all j. The value of τ captures the essential parallelism in the method, as it indicates the number of processors that are involved in the computation.

Algorithm 1 Asynchronous Stochastic Coordinate Descent Algorithm $x_{K+1} = \text{AsySCD}(x_0, \gamma, K)$

Require: $x_0 \in \Omega$, γ, and K
Ensure: x_{K+1}
1: Initialize $j \leftarrow 0$;
2: **while** $j \leq K$ **do**
3: Choose $i(j)$ from $\{1, \ldots, n\}$ with equal probability;
4: $x_{j+1} \leftarrow \mathcal{P}_\Omega \left(x_j - \frac{\gamma}{L_{\max}} e_{i(j)} \nabla_{i(j)} f(x_{k(j)}) \right)$;
5: $j \leftarrow j + 1$;
6: **end while**

The projection operation P_Ω onto the feasible set is not needed in the case of unconstrained optimization. For separable constraints (2), it requires a simple clipping operation on the $i(j)$ component of x.

We note several differences with earlier asynchronous approaches. Unlike the asynchronous scheme in Bertsekas and Tsitsiklis (1989, Section 6.1), the *latest* value of x is updated at each step, not an earlier iterate. Although our model of computation is similar to Hogwild! (Niu et al., 2011), the algorithm differs in that each iteration of AsySCD evaluates a single component of the gradient exactly, while Hogwild! computes only a (usually crude) estimate of the full gradient. Our analysis of AsySCD below is comprehensively different from that of Niu et al. (2011), and we obtain stronger convergence results.

4. Unconstrained Smooth Convex Case

This section presents results about convergence of AsySCD in the unconstrained case $\Omega = \mathbb{R}^n$. The theorem encompasses both the linear rate for essentially strongly convex f and the sublinear rate for general convex f. The result depends strongly on the delay parameter τ. (Proofs of results in this section appear in Appendix A.) In Algorithm 1, the indices $i(j)$, $j = 0, 1, 2, \ldots$ are random variables. We denote the expectation over all random variables as $\mathbb{E}$, the conditional expectation in term of $i(j)$ given $i(0), i(1), \cdots, i(j-1)$ as $\mathbb{E}_{i(j)}$.

A crucial issue in AsySCD is the choice of steplength parameter γ. This choice involves a tradeoff: We would like γ to be long enough that significant progress is made at each step, but not so long that the gradient information computed at step $k(j)$ is stale and irrelevant by the time the update is applied at step j. We enforce this tradeoff by means of a bound on the ratio of expected squared norms on ∇f at successive iterates; specifically,

$$\rho^{-1} \leq \frac{\mathbb{E}\|\nabla f(x_{j+1})\|^2}{\mathbb{E}\|\nabla f(x_j)\|^2} \leq \rho, \tag{4}$$

where $\rho > 1$ is a user defined parameter. The analysis becomes a delicate balancing act in the choice of ρ and steplength γ between aggression and excessive conservatism. We find, however, that these values can be chosen to ensure steady convergence for the asynchronous

method at a *linear* rate, with rate constants that are almost consistent with vanilla short-step full-gradient descent.

Theorem 1 *Suppose that $\Omega = \mathbb{R}^n$ in (1) and that Assumption 1 is satisfied. For any $\rho > 1$, define the quantity ψ as follows:*

$$\psi := 1 + \frac{2\tau\rho^\tau L_{\text{res}}}{\sqrt{n}L_{\max}}. \tag{5}$$

Suppose that the steplength parameter $\gamma > 0$ satisfies the following three upper bounds:

$$\gamma \leq \frac{1}{\psi}, \tag{6a}$$

$$\gamma \leq \frac{(\rho-1)\sqrt{n}L_{\max}}{2\rho^{\tau+1}L_{\text{res}}}, \tag{6b}$$

$$\gamma \leq \frac{(\rho-1)\sqrt{n}L_{\max}}{L_{\text{res}}\rho^\tau(2+\frac{L_{\text{res}}}{\sqrt{n}L_{\max}})}. \tag{6c}$$

Then we have that for any $j \geq 0$ that

$$\rho^{-1}\mathbb{E}(\|\nabla f(x_j)\|^2) \leq \mathbb{E}(\|\nabla f(x_{j+1})\|^2) \leq \rho\mathbb{E}(\|\nabla f(x_j)\|^2). \tag{7}$$

Moreover, if the essentially strong convexity property (3) holds with $l > 0$, we have

$$\mathbb{E}(f(x_j) - f^*) \leq \left(1 - \frac{2l\gamma}{nL_{\max}}\left(1 - \frac{\psi}{2}\gamma\right)\right)^j (f(x_0) - f^*), \tag{8}$$

while for general smooth convex functions f, we have

$$\mathbb{E}(f(x_j) - f^*) \leq \frac{1}{(f(x_0) - f^*)^{-1} + j\gamma(1-\frac{\psi}{2}\gamma)/(nL_{\max}R^2)}. \tag{9}$$

This theorem demonstrates linear convergence (8) for AsySCD in the unconstrained essentially strongly convex case. This result is better than that obtained for Hogwild! (Niu et al., 2011), which guarantees only sublinear convergence under the stronger assumption of strict convexity.

The following corollary proposes an interesting particular choice of the parameters for which the convergence expressions become more comprehensible. The result requires a condition on the delay bound τ in terms of n and the ratio $L_{\max}/L_{\text{res}}$.

Corollary 2 *Suppose that Assumption 1 holds, and that*

$$\tau + 1 \leq \frac{\sqrt{n}L_{\max}}{2eL_{\text{res}}}. \tag{10}$$

Then if we choose

$$\rho = 1 + \frac{2eL_{\text{res}}}{\sqrt{n}L_{\max}}, \tag{11}$$

define ψ by (5), and set $\gamma = 1/\psi$, we have for the essentially strongly convex case (3) with $l > 0$ that

$$\mathbb{E}(f(x_j) - f^*) \leq \left(1 - \frac{l}{2nL_{\max}}\right)^j (f(x_0) - f^*), \tag{12}$$

while for the case of general convex f, we have

$$\mathbb{E}(f(x_j) - f^*) \leq \frac{1}{(f(x_0) - f^*)^{-1} + j/(4nL_{\max}R^2)}. \tag{13}$$

We note that the linear rate (12) is broadly consistent with the linear rate for the classical steepest descent method applied to strongly convex functions, which has a rate constant of $(1-2l/L)$, where L is the standard Lipschitz constant for ∇f. If we assume (not unreasonably) that n steps of stochastic coordinate descent cost roughly the same as one step of steepest descent, and note from (12) that n steps of stochastic coordinate descent would achieve a reduction factor of about $(1 - l/(2L_{\max}))$, a standard argument would suggest that stochastic coordinate descent would require about $4L_{\max}/L$ times more computation. (Note that $L_{\max}/L \in [1/n, 1]$.) The stochastic approach may gain an advantage from the parallel implementation, however. Steepest descent requires synchronization and careful division of gradient evaluations, whereas the stochastic approach can be implemented in an asynchronous fashion.

For the general convex case, (13) defines a sublinear rate, whose relationship with the rate of the steepest descent for general convex optimization is similar to the previous paragraph.

As noted in Section 1, the parameter τ is closely related to the number of cores that can be involved in the computation, without degrading the convergence performance of the algorithm. In other words, if the number of cores is small enough such that (10) holds, the convergence expressions (12), (13) do not depend on the number of cores, implying that linear speedup can be expected. A small value for the ratio $L_{\text{res}}/L_{\max}$ (not much greater than 1) implies a greater degree of potential parallelism. As we note at the end of Section 1, this ratio tends to be small in some important applications — a situation that would allow $O(\sqrt{n})$ cores to be used with near-linear speedup.

We conclude this section with a high-probability estimate for convergence of the sequence of function values.

Theorem 3 *Suppose that the assumptions of Corollary 2 hold, including the definitions of ρ and ψ. Then for any $\epsilon \in (0, f(x_0) - f^*)$ and $\eta \in (0, 1)$, we have that*

$$\mathbb{P}\left(f(x_j) - f^* \leq \epsilon\right) \geq 1 - \eta, \tag{14}$$

provided that either of the following sufficient conditions hold for the index j. In the essentially strongly convex case (3) with $l > 0$, it suffices to have

$$j \geq \frac{2nL_{\max}}{l} \left|\log \frac{f(x_0) - f^*}{\epsilon\eta}\right|, \tag{15}$$

while in the general convex case, a sufficient condition is

$$j \geq 4nL_{\max}R^2 \left(\frac{1}{\epsilon\eta} - \frac{1}{f(x_0) - f^*}\right). \tag{16}$$

5. Constrained Smooth Convex Case

This section considers the case of separable constraints (2). We show results about convergence rates and high-probability complexity estimates, analogous to those of the previous section. Proofs appear in Appendix B.

As in the unconstrained case, the steplength γ should be chosen to ensure steady progress while ensuring that update information does not become too stale. Because constraints are present, the ratio (4) is no longer appropriate. We use instead a ratio of squares of expected differences in successive primal iterates:

$$\frac{\mathbb{E}\|x_{j-1} - \bar{x}_j\|^2}{\mathbb{E}\|x_j - \bar{x}_{j+1}\|^2}, \tag{17}$$

where $\bar{x}_{j+1}$ is the hypothesized full update obtained by applying the single-component update to *every* component of x_j, that is,

$$\bar{x}_{j+1} := \arg\min_{x \in \Omega} \langle \nabla f(x_{k(j)}), x - x_j \rangle + \frac{L_{\max}}{2\gamma} \|x - x_j\|^2.$$

In the unconstrained case $\Omega = \mathbb{R}^n$, the ratio (17) reduces to

$$\frac{\mathbb{E}\|\nabla f(x_{k(j-1)})\|^2}{\mathbb{E}\|\nabla f(x_{k(j)})\|^2},$$

which is evidently related to (4), but not identical.

We have the following result concerning convergence of the expected error to zero.

Theorem 4 *Suppose that Ω has the form (2), that Assumption 1 is satisfied, and that $n \geq 5$. Let ρ be a constant with $\rho > (1 - 2/\sqrt{n})^{-1}$, and define the quantity ψ as follows:*

$$\psi := 1 + \frac{L_{\text{res}} \tau \rho^\tau}{\sqrt{n} L_{\max}} \left(2 + \frac{L_{\max}}{\sqrt{n} L_{\text{res}}} + \frac{2\tau}{n} \right). \tag{18}$$

Suppose that the steplength parameter $\gamma > 0$ satisfies the following two upper bounds:

$$\gamma \leq \frac{1}{\psi}, \quad \gamma \leq \left(1 - \frac{1}{\rho} - \frac{2}{\sqrt{n}} \right) \frac{\sqrt{n} L_{\max}}{4 L_{\text{res}} \tau \rho^\tau}. \tag{19}$$

Then we have

$$\mathbb{E}\|x_{j-1} - \bar{x}_j\|^2 \leq \rho \mathbb{E}\|x_j - \bar{x}_{j+1}\|^2, \quad j = 1, 2, \ldots . \tag{20}$$

If the essential strong convexity property (3) holds with $l > 0$, we have for $j = 1, 2, \ldots$ that

$$\begin{aligned} &\mathbb{E}\|x_j - \mathcal{P}_S(x_j)\|^2 + \frac{2\gamma}{L_{\max}} (\mathbb{E} f(x_j) - f^*) \\ &\leq \left(1 - \frac{l}{n(l + \gamma^{-1} L_{\max})} \right)^j \left(R^2 + \frac{2\gamma}{L_{\max}} (f(x_0) - f^*) \right). \end{aligned} \tag{21}$$

For general smooth convex function f, we have

$$\mathbb{E} f(x_j) - f^* \leq \frac{n(R^2 L_{\max} + 2\gamma(f(x_0) - f^*))}{2\gamma(n + j)}. \tag{22}$$

Similarly to the unconstrained case, the following corollary proposes an interesting particular choice for the parameters for which the convergence expressions become more comprehensible. The result requires a condition on the delay bound τ in terms of n and the ratio $L_{\max}/L_{\text{res}}$.

Corollary 5 *Suppose that Assumption 1 holds, that $\tau \geq 1$ and $n \geq 5$, and that*

$$\tau(\tau+1) \leq \frac{\sqrt{n}L_{\max}}{4eL_{\text{res}}}. \tag{23}$$

If we choose

$$\rho = 1 + \frac{4e\tau L_{\text{res}}}{\sqrt{n}L_{\max}}, \tag{24}$$

then the steplength $\gamma = 1/2$ will satisfy the bounds (19). In addition, for the essentially strongly convex case (3) with $l > 0$, we have for $j = 1, 2, \ldots$ that

$$\mathbb{E}(f(x_j) - f^*) \leq \left(1 - \frac{l}{n(l + 2L_{\max})}\right)^j (L_{\max}R^2 + f(x_0) - f^*), \tag{25}$$

while for the case of general convex f, we have

$$\mathbb{E}(f(x_j) - f^*) \leq \frac{n(L_{\max}R^2 + f(x_0) - f^*)}{j + n}. \tag{26}$$

Similarly to Section 4, and provided τ satisfies (23), the convergence rate is not affected appreciably by the delay bound τ, and near-linear speedup can be expected for multicore implementations when (23) holds. This condition is more restrictive than (10) in the unconstrained case, but still holds in many problems for interesting values of τ. When $L_{\text{res}}/L_{\max}$ is bounded independently of dimension, the maximal number of cores allowed is of the the order of $n^{1/4}$, which is smaller than the $O(n^{1/2})$ value obtained for the unconstrained case.

We conclude this section with another high-probability bound, whose proof tracks that of Theorem 3.

Theorem 6 *Suppose that the conditions of Corollary 5 hold, including the definitions of ρ and ψ. Then for $\epsilon > 0$ and $\eta \in (0, 1)$, we have that*

$$\mathbb{P}\left(f(x_j) - f^* \leq \epsilon\right) \geq 1 - \eta,$$

provided that one of the following conditions holds: In the essentially strongly convex case (3) with $l > 0$, we require

$$j \geq \frac{n(l + 2L_{\max})}{l} \left|\log \frac{L_{\max}R^2 + f(x_0) - f^*}{\epsilon\eta}\right|,$$

while in the general convex case, it suffices that

$$j \geq \frac{n(L_{\max}R^2 + f(x_0) - f^*)}{\epsilon\eta} - n.$$

6. Experiments

We illustrate the behavior of two variants of the stochastic coordinate descent approach on test problems constructed from several data sets. Our interests are in the efficiency of multicore implementations (by comparison with a single-threaded implementation) and in performance relative to alternative solvers for the same problems.

All our test problems have the form (1), with either $\Omega = \mathbb{R}^n$ or Ω separable as in (2). The objective f is quadratic, that is,

$$f(x) = \frac{1}{2}x^T Q x + c^T x,$$

with Q symmetric positive definite.

Our implementation of AsySCD is called DIMM-WITTED (or DW for short). It runs on various numbers of threads, from 1 to 40, each thread assigned to a single core in our 40-core Intel Xeon architecture. Cores on the Xeon architecture are arranged into four sockets — ten cores per socket, with each socket having its own memory. Non-uniform memory access (NUMA) means that memory accesses to local memory (on the same socket as the core) are less expensive than accesses to memory on another socket. In our DW implementation, we assign each socket an equal-sized "slice" of Q, a row submatrix. The components of x are partitioned between cores, each core being responsible for updating its own partition of x (though it can read the components of x from other cores). The components of x assigned to the cores correspond to the rows of Q assigned to that core's socket. Computation is grouped into "epochs," where an epoch is defined to be the period of computation during which each component of x is updated exactly once. We use the parameter p to denote the number of epochs that are executed between reordering (shuffling) of the coordinates of x. We investigate both shuffling after every epoch ($p = 1$) and after every tenth epoch ($p = 10$). Access to x is lock-free, and updates are performed asynchronously. This update scheme does not implement exactly the "sampling with replacement" scheme analyzed in previous sections, but can be viewed as a high performance, practical adaptation of the AsySCD method.

To do each coordinate descent update, a thread must read the latest value of x. Most components are already in the cache for that core, so that it only needs to fetch those components recently changed. When a thread writes to x_i, the hardware ensures that this x_i is simultaneously removed from other cores, signaling that they must fetch the updated version before proceeding with their respective computations.

Although DW is not a precise implementation of AsySCD, it largely achieves the consistent-read condition that is assumed by the analysis. Inconsistent read happens on a core only if the following three conditions are satisfied simultaneously:

- A core does not finish reading recently changed coordinates of x (note that it needs to read no more than τ coordinates);
- Among these recently changed coordinates, modifications take place both to coordinates that *have been read* and that are *still to be read* by this core;
- Modification of the already-read coordinates happens earlier than the modification of the still-unread coordinates.

Inconsistent read will occur only if at least two coordinates of x are modified twice during a stretch of approximately τ updates to x (that is, iterations of Algorithm 1). For the DW implementation, inconsistent read would require repeated updating of a particular component in a stretch of approximately τ iterations that straddles two epochs. This event would be rare, for typical values of n and τ. Of course, one can avoid the inconsistent read issue altogether by changing the shuffling rule slightly, enforcing the requirement that no coordinate can be modified twice in a span of τ iterations. From the practical perspective, this change does not improve performance, and detracts from the simplicity of the approach. From the theoretical perspective, however, the analysis for the inconsistent-read model would be interesting and meaningful, and we plan to study this topic in future work.

The first test problem `QP` is an unconstrained, regularized least squares problem constructed with synthetic data. It has the form

$$\min_{x\in\mathbb{R}^n} \; f(x) := \frac{1}{2}\|Ax-b\|^2 + \frac{\alpha}{2}\|x\|^2. \tag{27}$$

All elements of $A \in \mathbb{R}^{m\times n}$, the true model $\tilde{x} \in \mathbb{R}^n$, and the observation noise vector $\delta \in \mathbb{R}^m$ are generated in i.i.d. fashion from the Gaussian distribution $\mathcal{N}(0,1)$, following which each column in A is scaled to have a Euclidean norm of 1. The observation $b \in \mathbb{R}^m$ is constructed from $A\tilde{x} + \delta\|A\tilde{x}\|/(5m)$. We choose $m = 6000$, $n = 20000$, and $\alpha = 0.5$. We therefore have $L_{\max} = 1 + \alpha = 1.5$ and

$$\frac{L_{\text{res}}}{L_{\max}} \approx \frac{1+\sqrt{n/m}+\alpha}{1+\alpha} \approx 2.2.$$

This problem is diagonally dominant, and the condition (10) is satisfied when delay parameter τ is less than about 95. In Algorithm 1, we set the steplength parameter γ to 1, and we choose initial iterate to be $x_0 = \mathbf{0}$. We measure convergence of the residual norm $\|\nabla f(x)\|$.

Our second problem `QPc` is a bound-constrained version of (27):

$$\min_{x\in\mathbb{R}^n_+} \quad f(x) := \frac{1}{2}(x-\tilde{x})^T(A^TA+\alpha I)(x-\tilde{x}). \tag{28}$$

The methodology for generating A and $\tilde{x}$ and for choosing the values of m, n, γ, and x_0 is the same as for (27). We measure convergence via the residual $\|x - \mathcal{P}_\Omega(x - \nabla f(x))\|$, where Ω is the nonnegative orthant $\mathbb{R}^n_+$. At the solution of (28), about half the components of x are at their lower bound of 0.

Our third and fourth problems are quadratic penalty functions for linear programming relaxations of vertex cover problems on large graphs. The vertex cover problem for an undirected graph with edge set E and vertex set V can be written as a binary linear program:

$$\min_{y\in\{0,1\}^{|V|}} \sum_{v\in V} y_v \quad \text{subject to } \; y_u + y_v \geq 0, \quad \forall\,(u,v) \in E.$$

By relaxing each binary constraint to the interval $[0,1]$, introducing slack variables for the cover inequalities, we obtain a problem of the form

$$\min_{y_v\in[0,1],\, s_{uv}\in[0,1]} \sum_{v\in V} y_v \quad \text{subject to } \; y_u + y_v - s_{uv} = 0, \quad \forall\,(u,v) \in E.$$

This has the form

$$\min_{x\in[0,1]^n} c^T x \quad \text{subject to } Ax = b,$$

for $n = |V| + |E|$. The test problem (29) is a regularized quadratic penalty reformulation of this linear program for some penalty parameter β:

$$\min_{x\in[0,1]^n} \quad c^T x + \frac{\beta}{2}\|Ax - b\|^2 + \frac{1}{2\beta}\|x\|^2, \tag{29}$$

with $\beta = 5$. Two test data sets `Amazon` and `DBLP` have dimensions $n = 561050$ and $n = 520891$, respectively.

We tracked the behavior of the residual as a function of the number of epochs, when executed on different numbers of cores. Figure 1 shows convergence behavior for each of our four test problems on various numbers of cores with two different shuffling periods: $p = 1$ and $p = 10$. We note the following points.

- The total amount of computation to achieve any level of precision appears to be almost independent of the number of cores, at least up to 40 cores. In this respect, the performance of the algorithm does not change appreciably as the number of cores is increased. Thus, any deviation from linear speedup is due not to degradation of convergence speed in the algorithm but rather to systems issues in the implementation.

- When we reshuffle after every epoch ($p = 1$), convergence is slightly faster in synthetic unconstrained `QP` but slightly slower in `Amazon` and `DBLP` than when we do occasional reshuffling ($p = 10$). Overall, the convergence rates with different shuffling periods are comparable in the sense of epochs. However, when the dimension of the variable is large, the shuffling operation becomes expensive, so we would recommend using a large value for p for large-dimensional problems.

Results for speedup on multicore implementations are shown in Figures 2 and 3 for DW with $p = 10$. Speedup is defined as follows:

$$\frac{\text{runtime a single core using DW}}{\text{runtime on } P \text{ cores}}.$$

Near-linear speedup can be observed for the two QP problems with synthetic data. For Problems 3 and 4, speedup is at most 12-14; there are few gains when the number of cores exceeds about 12. We believe that the degradation is due mostly to memory contention. Although these problems have high dimension, the matrix Q is very sparse (in contrast to the dense Q for the synthetic data set). Thus, the ratio of computation to data movement / memory access is much lower for these problems, making memory contention effects more significant.

Figures 2 and 3 also show results of a global-locking strategy for the parallel stochastic coordinate descent method, in which the vector x is locked by a core whenever it performs a read or update. The performance curve for this strategy hugs the horizontal axis; it is not competitive.

Wall clock times required for the four test problems on 1 and 40 cores, to reduce residuals below 10^{-5} are shown in Table 1. (Similar speedups are noted when we use a convergence tolerance looser than 10^{-5}.)

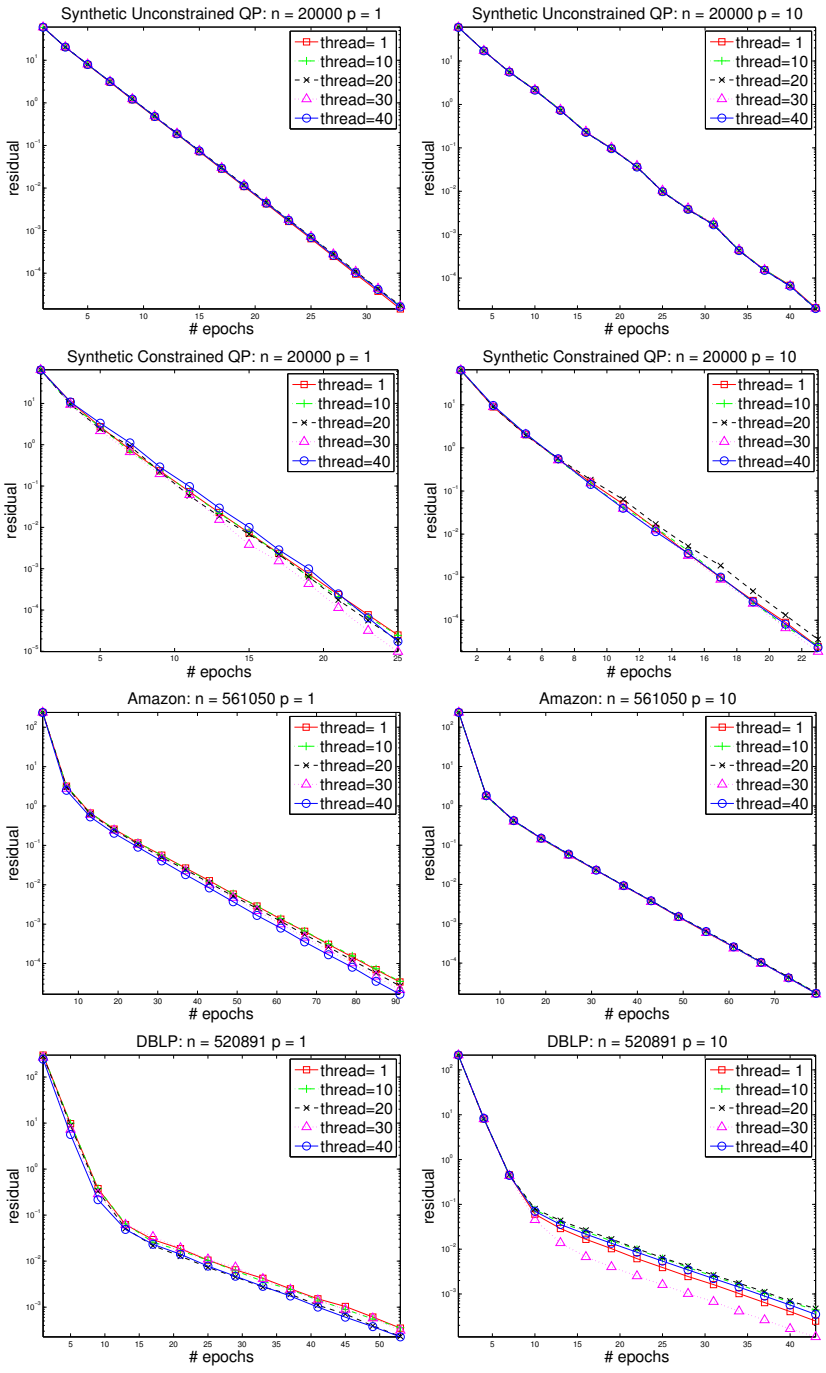


Figure 1: Residuals vs epoch number for the four test problems. Results are reported for variants in which indices are reshuffled after every epoch ($p = 1$) and after every tenth epoch ($p = 10$).

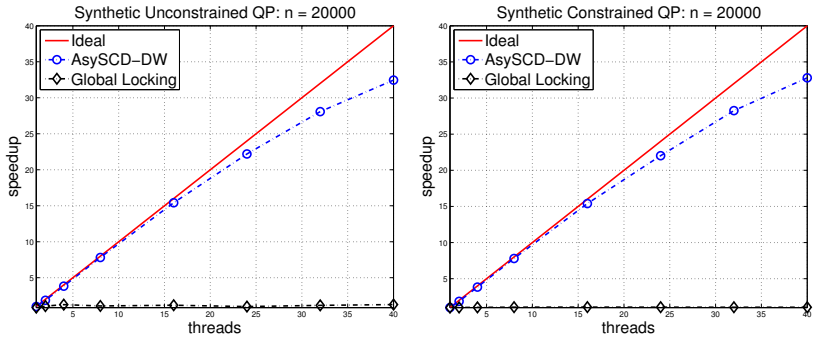


Figure 2: Test problems 1 and 2: Speedup of multicore implementations of DW on up to 40 cores of an Intel Xeon architecture. Ideal (linear) speedup curve is shown for reference, along with poor speedups obtained for a global-locking strategy.

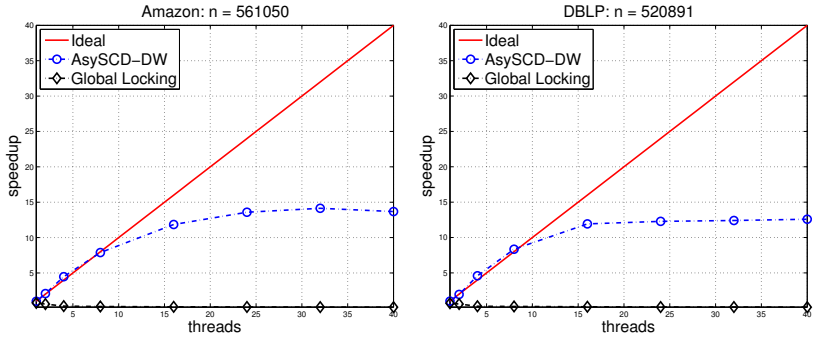


Figure 3: Test problems 3 and 4: Speedup of multicore implementations of DW on up to 40 cores of an Intel Xeon architecture. Ideal (linear) speedup curve is shown for reference, along with poor speedups obtained for a global-locking strategy.

Problem	1 core	40 cores
`QP`	98.4	3.03
`QPc`	59.7	1.82
`Amazon`	17.1	1.25
`DBLP`	11.5	.91

Table 1: Runtimes (seconds) for the four test problems on 1 and 40 cores.

All problems reported on above are essentially strongly convex. Similar speedup properties can be obtained in the weakly convex case as well. We show speedups for the `QPc` problem with $\alpha = 0$. Table 2 demonstrates similar speedup to the essentially strongly convex case shown in Figure 2.

Turning now to comparisons between AsySCD and alternative algorithms, we start by considering the basic gradient descent method. We implement gradient descent in a

#cores	Time(sec)	Speedup
1	55.9	1
10	5.19	10.8
20	2.77	20.2
30	2.06	27.2
40	1.81	30.9

Table 2: Runtimes (seconds) and speedup for multicore implementations of DW on different number of cores for the weakly convex `QPc` problem (with $\alpha = 0$) to achieve a residual below 0.06.

#cores	Time(sec) SynGD / AsySCD	Speedup SynGD / AsySCD
1	121. / 27.1	0.22 / 1.00
10	11.4 / 2.57	2.38 / 10.5
20	6.00 / 1.36	4.51 / 19.9
30	4.44 / 1.01	6.10 / 26.8
40	3.91 / 0.88	6.93 / 30.8

Table 3: Efficiency comparison between SynGD and AsySCD for the `QP` problem. The running time and speedup are based on the residual achieving a tolerance of 10^{-5}.

Dataset	# of Samples	# of Features	Train time(sec) LIBSVM	Train time(sec) AsySCD
adult	32561	123	16.15	1.39
news	19996	1355191	214.48	7.22
rcv	20242	47236	40.33	16.06
reuters	8293	18930	1.63	0.81
w8a	49749	300	33.62	5.86

Table 4: Efficiency comparison between LIBSVM and AsySCD for kernel SVM using 40 cores using homogeneous kernels ($K(x_i, x_j) = (x_i^T x_j)^2$). The running time and speedup are calculated based on the "residual" 10^{-3}. Here, to make both algorithms comparable, the "residual" is defined by $\|x - \mathcal{P}_\Omega(x - \nabla f(x))\|_\infty$.

parallel, synchronous fashion, distributing the gradient computation load on multiple cores and updates the variable x in parallel at each step. The resulting implementation is called SynGD. Table 3 reports running time and speedup of both AsySCD over SynGD, showing a clear advantage for AsySCD.

Next we compare AsySCD to LIBSVM (Chang and Lin, 2011) a popular parallel solver for kernel support vector machines (SVM). Both algorithms are run on 40 cores to solve the dual formulation of kernel SVM, without an intercept term. All data sets used in 4 except

reuters were obtained from the LIBSVM data set repository.[1] The data set reuters is a sparse binary text classification data set constructed as a one-versus-all version of Reuters-2159.[2] Our comparisons, shown in Table 4, indicate that AsySCD outperforms LIBSVM on these test sets.

7. Extension

The AsySCD algorithm can be extended by partitioning the coordinates into blocks, and modifying Algorithm 1 to work with these blocks rather than with single coordinates. If L_i, $L_{\max}$, and L_{res} are defined in the block sense, as follows:

$$\|\nabla f(x) - \nabla f(x + E_i t)\| \leq L_{\text{res}}\|t\| \quad \forall x, i, t \in \mathbb{R}^{|i|},$$
$$\|\nabla_i f(x) - \nabla_i f(x + E_i t)\| \leq L_i\|t\| \quad \forall x, i, t \in \mathbb{R}^{|i|},$$
$$L_{\max} = \max_i L_i,$$

where E_i is the projection from the ith block to $\mathbb{R}^n$ and $|i|$ denotes the number of components in block i, our analysis can be extended appropriately.

To make the AsySCD algorithm more efficient, one can redefine the steplength in Algorithm 1 to be $\frac{\gamma}{L_{i(j)}}$ rather than $\frac{\gamma}{L_{\max}}$. Our analysis can be applied to this variant by doing a change of variables to $\tilde{x}$, with $x_i = \frac{L_i}{L_{\max}}\tilde{x}_i$ and defining L_i, L_{res}, and $L_{\max}$ in terms of $\tilde{x}$.

8. Conclusion

This paper proposes an asynchronous parallel stochastic coordinate descent algorithm for minimizing convex objectives, in the unconstrained and separable-constrained cases. Sublinear convergence (at rate $1/K$) is proved for general convex functions, with stronger linear convergence results for functions that satisfy an essential strong convexity property. Our analysis indicates the extent to which parallel implementations can be expected to yield near-linear speedup, in terms of a parameter that quantifies the cross-coordinate interactions in the gradient ∇f and a parameter τ that bounds the delay in updating. Our computational experience confirms the theory.

Acknowledgments

This project is supported by NSF Grants DMS-0914524, DMS-1216318, and CCF-1356918; NSF CAREER Award IIS-1353606; ONR Awards N00014-13-1-0129 and N00014-12-1-0041; AFOSR Award FA9550-13-1-0138; a Sloan Research Fellowship; and grants from Oracle, Google, and ExxonMobil.

Appendix A. Proofs for Unconstrained Case

This section contains convergence proofs for AsySCD in the unconstrained case.

1. http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/
2. http://www.daviddlewis.com/resources/testcollections/reuters21578/

We start with a technical result, then move to the proofs of the three main results of Section 4.

Lemma 7 *For any x, we have*

$$\|x - \mathcal{P}_S(x)\|^2 \|\nabla f(x)\|^2 \geq (f(x) - f^*)^2.$$

If the essential strong convexity property (3) holds, we have

$$\|\nabla f(x)\|^2 \geq 2l(f(x) - f^*).$$

Proof The first inequality is proved as follows:

$$f(x) - f^* \leq \langle \nabla f(x), x - \mathcal{P}_S(x)\rangle \leq \|\nabla f(x)\| \|\mathcal{P}_S(x) - x\|.$$

For the second bound, we have from the definition (3), setting $y \leftarrow x$ and $x \leftarrow \mathcal{P}_S(x)$, that

$$\begin{aligned} f^* - f(x) &\geq \langle \nabla f(x), \mathcal{P}_S(x) - x\rangle + \frac{l}{2}\|x - \mathcal{P}_S(x)\|^2 \\ &= \frac{l}{2}\|\mathcal{P}_S(x) - x + \frac{1}{l}\nabla f(x)\|^2 - \frac{1}{2l}\|\nabla f(x)\|^2 \geq -\frac{1}{2l}\|\nabla f(x)\|^2, \end{aligned}$$

as required. ■

Proof (Theorem 1) We prove each of the two inequalities in (7) by induction. We start with the left-hand inequality. For all values of j, we have

$$\begin{aligned} &\mathbb{E}\left(\|\nabla f(x_j)\|^2 - \|\nabla f(x_{j+1})\|^2\right) \\ &\quad = \mathbb{E}\langle \nabla f(x_j) + \nabla f(x_{j+1}), \nabla f(x_j) - \nabla f(x_{j+1})\rangle \\ &\quad = \mathbb{E}\langle 2\nabla f(x_j) + \nabla f(x_{j+1}) - \nabla f(x_j), \nabla f(x_j) - \nabla f(x_{j+1})\rangle \\ &\quad \leq 2\mathbb{E}\langle \nabla f(x_j), \nabla f(x_j) - \nabla f(x_{j+1})\rangle \\ &\quad \leq 2\mathbb{E}(\|\nabla f(x_j)\| \|\nabla f(x_j) - \nabla f(x_{j+1})\|) \\ &\quad \leq 2L_{\rm res}\mathbb{E}(\|\nabla f(x_j)\| \|x_j - x_{j+1}\|) \\ &\quad \leq \frac{2L_{\rm res}\gamma}{L_{\max}}\mathbb{E}(\|\nabla f(x_j)\| \|\nabla_{i(j)} f(x_{k(j)})\|) \\ &\quad \leq \frac{L_{\rm res}\gamma}{L_{\max}}\mathbb{E}(n^{-1/2}\|\nabla f(x_j)\|^2 + n^{1/2}\|\nabla_{i(j)} f(x_{k(j)})\|^2) \\ &\quad = \frac{L_{\rm res}\gamma}{L_{\max}}\mathbb{E}(n^{-1/2}\|\nabla f(x_j)\|^2 + n^{1/2}\mathbb{E}_{i(j)}(\|\nabla_{i(j)} f(x_{k(j)})\|^2)) \\ &\quad = \frac{L_{\rm res}\gamma}{L_{\max}}\mathbb{E}(n^{-1/2}\|\nabla f(x_j)\|^2 + n^{-1/2}\|\nabla f(x_{k(j)})\|^2) \\ &\quad \leq \frac{L_{\rm res}\gamma}{\sqrt{n}L_{\max}}\mathbb{E}\left(\|\nabla f(x_j)\|^2 + \|\nabla f(x_{k(j)})\|^2\right). \end{aligned} \tag{30}$$

We can use this bound to show that the left-hand inequality in (7) holds for $j = 0$. By setting $j = 0$ in (30) and noting that $k(0) = 0$, we obtain

$$\mathbb{E}\left(\|\nabla f(x_0)\|^2 - \|\nabla f(x_1)\|^2\right) \leq \frac{L_{\rm res}\gamma}{\sqrt{n}L_{\max}} 2\mathbb{E}(\|\nabla f(x_0)\|^2). \tag{31}$$

From (6b), we have

$$\frac{2L_{\rm res}\gamma}{\sqrt{n}L_{\max}} \leq \frac{\rho-1}{\rho^\tau} \leq \frac{\rho-1}{\rho} = 1-\rho^{-1},$$

where the second inequality follows from $\rho > 1$. By substituting into (31), we obtain $\rho^{-1}\mathbb{E}(\|\nabla f(x_0)\|^2) \leq \mathbb{E}(\|\nabla f(x_1)\|^2)$, establishing the result for $j = 1$. For the inductive step, we use (30) again, assuming that the left-hand inequality in (7) holds up to stage j, and thus that

$$\mathbb{E}(\|\nabla f(x_{k(j)})\|^2) \leq \rho^\tau \mathbb{E}(\|\nabla f(x_j)\|^2),$$

provided that $0 \leq j - k(j) \leq \tau$, as assumed. By substituting into the right-hand side of (30) again, and using $\rho > 1$, we obtain

$$\mathbb{E}\left(\|\nabla f(x_j)\|^2 - \|\nabla f(x_{j+1})\|^2\right) \leq \frac{2L_{\rm res}\gamma\rho^\tau}{\sqrt{n}L_{\max}}\mathbb{E}\left(\|\nabla f(x_j)\|^2\right).$$

By substituting (6b) we conclude that the left-hand inequality in (7) holds for all j.

We now work on the right-hand inequality in (7). For all j, we have the following:

$$\begin{aligned}
&\mathbb{E}\left(\|\nabla f(x_{j+1})\|^2 - \|\nabla f(x_j)\|^2\right)\\
&\quad= \mathbb{E}\langle \nabla f(x_j) + \nabla f(x_{j+1}), \nabla f(x_{j+1}) - \nabla f(x_j)\rangle\\
&\quad\leq \mathbb{E}(\|\nabla f(x_j) + \nabla f(x_{j+1})\|\|\nabla f(x_j) - \nabla f(x_{j+1})\|)\\
&\quad\leq L_{\rm res}\mathbb{E}(\|\nabla f(x_j) + \nabla f(x_{j+1})\|\|x_j - x_{j+1}\|)\\
&\quad\leq L_{\rm res}\mathbb{E}((2\|\nabla f(x_j)\| + \|\nabla f(x_{j+1}) - \nabla f(x_j)\|)\|x_j - x_{j+1}\|)\\
&\quad\leq L_{\rm res}\mathbb{E}(2\|\nabla f(x_j)\|\|x_j - x_{j+1}\| + L_{\rm res}\|x_j - x_{j+1}\|^2)\\
&\quad\leq L_{\rm res}\mathbb{E}\left(\frac{2\gamma}{L_{\max}}\|\nabla f(x_j)\|\|\nabla_{i(j)} f(x_{k(j)})\| + \frac{L_{\rm res}\gamma^2}{L_{\max}^2}\|\nabla_{i(j)} f(x_{k(j)})\|^2\right)\\
&\quad\leq L_{\rm res}\mathbb{E}\left(\frac{\gamma}{L_{\max}}(n^{-1/2}\|\nabla f(x_j)\|^2 + n^{1/2}\|\nabla_{i(j)} f(x_{k(j)})\|^2 + \frac{L_{\rm res}\gamma^2}{L_{\max}^2}\|\nabla_{i(j)} f(x_{k(j)})\|^2\right)\\
&\quad= L_{\rm res}\mathbb{E}\left(\frac{\gamma}{L_{\max}}(n^{-1/2}\|\nabla f(x_j)\|^2 + n^{1/2}\mathbb{E}_{i(j)}(\|\nabla_{i(j)} f(x_{k(j)})\|^2)) + \right.\\
&\qquad\left.\frac{L_{\rm res}\gamma^2}{L_{\max}^2}\mathbb{E}_{i(j)}(\|\nabla_{i(j)} f(x_{k(j)})\|^2)\right)\\
&\quad= L_{\rm res}\mathbb{E}\left(\frac{\gamma}{L_{\max}}(n^{-1/2}\|\nabla f(x_j)\|^2 + n^{-1/2}\|\nabla f(x_{k(j)})\|^2) + \frac{L_{\rm res}\gamma^2}{nL_{\max}^2}\|\nabla f(x_{k(j)})\|^2\right)\\
&\quad= \frac{\gamma L_{\rm res}}{\sqrt{n}L_{\max}}\mathbb{E}\left(\|\nabla f(x_j)\|^2 + \|\nabla f(x_{k(j)})\|^2\right) + \frac{\gamma^2 L_{\rm res}^2}{nL_{\max}^2}\mathbb{E}(\|\nabla f(x_{k(j)})\|^2)\\
&\quad\leq \frac{\gamma L_{\rm res}}{\sqrt{n}L_{\max}}\mathbb{E}(\|\nabla f(x_j)\|^2) + \left(\frac{\gamma L_{\rm res}}{\sqrt{n}L_{\max}} + \frac{\gamma L_{\rm res}^2}{nL_{\max}^2}\right)\mathbb{E}(\|\nabla f(x_{k(j)})\|^2), \qquad (32)
\end{aligned}$$

where the last inequality is from the observation $\gamma \leq 1$. By setting $j = 0$ in this bound, and noting that $k(0) = 0$, we obtain

$$\mathbb{E}\left(\|\nabla f(x_1)\|^2 - \|\nabla f(x_0)\|^2\right) \leq \left(\frac{2\gamma L_{\rm res}}{\sqrt{n}L_{\max}} + \frac{\gamma L_{\rm res}^2}{nL_{\max}^2}\right)\mathbb{E}(\|\nabla f(x_0)\|^2). \qquad (33)$$

By using (6c), we have

$$\frac{2\gamma L_{\rm res}}{\sqrt{n}L_{\max}} + \frac{\gamma L_{\rm res}^2}{nL_{\max}^2} = \frac{L_{\rm res}\gamma}{\sqrt{n}L_{\max}}\left(2 + \frac{L_{\rm res}}{\sqrt{n}L_{\max}}\right) \leq \frac{\rho-1}{\rho^\tau} < \rho - 1,$$

where the last inequality follows from $\rho > 1$. By substituting into (33), we obtain $\mathbb{E}(\|\nabla f(x_1)\|^2) \leq \rho\mathbb{E}(\|\nabla f(x_0)\|^2)$, so the right-hand bound in (7) is established for $j = 0$. For the inductive step, we use (32) again, assuming that the right-hand inequality in (7) holds up to stage j, and thus that

$$\mathbb{E}(\|\nabla f(x_j)\|^2) \leq \rho^\tau \mathbb{E}(\|\nabla f(x_{k(j)})\|^2),$$

provided that $0 \leq j - k(j) \leq \tau$, as assumed. From (32) and the left-hand inequality in (7), we have by substituting this bound that

$$\mathbb{E}\left(\|\nabla f(x_{j+1})\|^2 - \|\nabla f(x_j)\|^2\right) \leq \left(\frac{2\gamma L_{\rm res}\rho^\tau}{\sqrt{n}L_{\max}} + \frac{\gamma L_{\rm res}^2\rho^\tau}{nL_{\max}^2}\right)\mathbb{E}(\|\nabla f(x_j)\|^2). \tag{34}$$

It follows immediately from (6c) that the term in parentheses in (34) is bounded above by $\rho - 1$. By substituting this bound into (34), we obtain $\mathbb{E}(\|\nabla f(x_{j+1})\|^2) \leq \rho\mathbb{E}(\|\nabla f(x_j)\|^2)$, as required.

At this point, we have shown that both inequalities in (7) are satisfied for all j.

Next we prove (8) and (9). Take the expectation of $f(x_{j+1})$ in terms of $i(j)$:

$$\begin{aligned}
\mathbb{E}_{i(j)} f(x_{j+1}) &= \mathbb{E}_{i(j)} f\left(x_j - \frac{\gamma}{L_{\max}} e_{i(j)} \nabla_{i(j)} f(x_{k(j)})\right) \\
&= \frac{1}{n}\sum_{i=1}^n f\left(x_j - \frac{\gamma}{L_{\max}} e_i \nabla_i f(x_{k(j)})\right) \\
&\leq \frac{1}{n}\sum_{i=1}^n f(x_j) - \frac{\gamma}{L_{\max}}\langle \nabla f(x_j), e_i \nabla_i f(x_{k(j)})\rangle + \frac{L_i}{2L_{\max}^2}\gamma^2 \|\nabla_i f(x_{k(j)})\|^2 \\
&\leq f(x_j) - \frac{\gamma}{nL_{\max}}\langle \nabla f(x_j), \nabla f(x_{k(j)})\rangle + \frac{\gamma^2}{2nL_{\max}}\|\nabla f(x_{k(j)})\|^2 \\
&= f(x_j) + \frac{\gamma}{nL_{\max}} \underbrace{\langle \nabla f(x_{k(j)}) - \nabla f(x_j), \nabla f(x_{k(j)})\rangle}_{T_1} \\
&\quad - \left(\frac{\gamma}{nL_{\max}} - \frac{\gamma^2}{2nL_{\max}}\right)\|\nabla f(x_{k(j)})\|^2.
\end{aligned} \tag{35}$$

The second term T_1 is caused by delay. If there is no the delay issue, T_1 should be 0 because of $\nabla f(x_j) = \nabla f(x_{k(j)})$. We estimate the upper bound of $\|\nabla f(x_{k(j)}) - \nabla f(x_j)\|$:

$$\begin{aligned}\|\nabla f(x_{k(j)}) - \nabla f(x_j)\| &\leq \sum_{d=k(j)}^{j-1} \|\nabla f(x_{d+1}) - \nabla f(x_d)\| \\ &\leq L_{\text{res}} \sum_{d=k(j)}^{j-1} \|x_{d+1} - x_d\| \\ &= \frac{L_{\text{res}}\gamma}{L_{\max}} \sum_{d=k(j)}^{j-1} \left\|\nabla_{i(d)} f(x_{k(d)})\right\|. \end{aligned} \tag{36}$$

Then $\mathbb{E}(|T_1|)$ can be bounded by

$$\begin{aligned}\mathbb{E}(|T_1|) &\leq \mathbb{E}(\|\nabla f(x_{k(j)}) - \nabla f(x_j)\|\|\nabla f(x_{k(j)})\|) \\ &\leq \frac{L_{\text{res}}\gamma}{L_{\max}} \mathbb{E}\left(\sum_{d=k(j)}^{j-1} \|\nabla_{i(d)} f(x_{k(d)})\|\|\nabla f(x_{k(j)})\|\right) \\ &\leq \frac{L_{\text{res}}\gamma}{2L_{\max}} \mathbb{E}\left(\sum_{d=k(j)}^{j-1} n^{1/2}\|\nabla_{i(d)} f(x_{k(d)})\|^2 + n^{-1/2}\|\nabla f(x_{k(j)})\|^2\right) \\ &= \frac{L_{\text{res}}\gamma}{2L_{\max}} \mathbb{E}\left(\sum_{d=k(j)}^{j-1} n^{1/2}\mathbb{E}_{i(d)}(\|\nabla_{i(d)} f(x_{k(d)})\|^2) + n^{-1/2}\|\nabla f(x_{k(j)})\|^2\right) \\ &= \frac{L_{\text{res}}\gamma}{2L_{\max}} \mathbb{E}\left(\sum_{d=k(j)}^{j-1} n^{-1/2}\|\nabla f(x_{k(d)})\|^2 + n^{-1/2}\|\nabla f(x_{k(j)})\|^2\right) \\ &= \frac{L_{\text{res}}\gamma}{2\sqrt{n}L_{\max}} \sum_{d=k(j)}^{j-1} \mathbb{E}(\|\nabla f(x_{k(d)})\|^2 + \|\nabla f(x_{k(j)})\|^2) \\ &\leq \frac{\tau\rho^\tau L_{\text{res}}\gamma}{\sqrt{n}L_{\max}} \mathbb{E}(\|\nabla f(x_{k(j)})\|^2) \end{aligned} \tag{37}$$

where the second line uses (36), and the final inequality uses the fact for d between $k(j)$ and $j-1$, $k(d)$ lies in the range $k(j) - \tau$ and $j-1$, so we have $|k(d) - k(j)| \leq \tau$ for all d.

Taking expectation on both sides of (35) in terms of all random variables, together with (37), we obtain

$$\begin{aligned}&\mathbb{E}(f(x_{j+1}) - f^*) \\ &\leq \mathbb{E}(f(x_j) - f^*) + \frac{\gamma}{nL_{\max}}\mathbb{E}(|T_1|) - \left(\frac{\gamma}{nL_{\max}} - \frac{\gamma^2}{2nL_{\max}}\right)\mathbb{E}(\|\nabla f(x_{k(j)})\|^2) \\ &\leq \mathbb{E}(f(x_j) - f^*) - \left(\frac{\gamma}{nL_{\max}} - \frac{\tau\rho^\tau L_{\text{res}}\gamma^2}{n^{3/2}L_{\max}^2} - \frac{\gamma^2}{2nL_{\max}}\right)\mathbb{E}(\|\nabla f(x_{k(j)})\|^2) \\ &= \mathbb{E}(f(x_j) - f^*) - \frac{\gamma}{nL_{\max}}\left(1 - \frac{\psi}{2}\gamma\right)\mathbb{E}(\|\nabla f(x_{k(j)})\|^2), \end{aligned}$$

which (because of (6a)) implies that $\mathbb{E}(f(x_j) - f^*)$ is monotonically decreasing. From Lemma 7 and the assumption $\|x_j - \mathcal{P}_S(x_j)\| \leq R$ for all j, we have

$$\begin{aligned}\|\nabla f(x_{k(j)})\|^2 &\geq \max\left\{2l(f(x_{k(j)}) - f^*),\ \frac{(f(x_{k(j)}) - f^*)^2}{\|x_{k(j)} - \mathcal{P}_S(x_{k(j)})\|^2}\right\} \\ &\geq \max\left\{2l(f(x_{k(j)}) - f^*),\ \frac{(f(x_{k(j)}) - f^*)^2}{R^2}\right\},\end{aligned}$$

which implies

$$\begin{aligned}\mathbb{E}(\|\nabla f(x_{k(j)})\|^2) &\geq \max\left\{2l\mathbb{E}(f(x_{k(j)} - f^*),\ \frac{\mathbb{E}(f(x_{k(j)} - f^*)^2}{R^2}\right\} \\ &\geq \max\left\{2l\mathbb{E}(f(x_j) - f^*),\ \frac{\mathbb{E}(f(x_j) - f^*)^2}{R^2}\right\}.\end{aligned}$$

From the first upper bound $\|\nabla f(x_{k(j)})\|^2 \geq 2l\mathbb{E}(f(x_j) - f^*)$, we have

$$\begin{aligned}\mathbb{E}(f(x_{j+1}) - f^*) &\leq \mathbb{E}(f(x_j) - f^*) - \frac{\gamma}{nL_{\max}}\left(1 - \frac{\psi}{2}\gamma\right)\mathbb{E}(\|\nabla f(x_{k(j)})\|^2) \\ &\leq \left(1 - \frac{2l\gamma}{nL_{\max}}\left(1 - \frac{\psi}{2}\gamma\right)\right)\mathbb{E}(f(x_j) - f^*),\end{aligned}$$

form which the linear convergence claim (8) follows by an obvious induction. From the other bound $\|\nabla f(x_{k(j)})\|^2 \geq \frac{(f(x_{k(j)}) - f^*)^2}{R^2}$, we have

$$\begin{aligned}\mathbb{E}(f(x_{j+1}) - f^*) &\leq \mathbb{E}(f(x_j) - f^*) - \frac{\gamma}{nL_{\max}}\left(1 - \frac{\psi}{2}\gamma\right)\mathbb{E}(\|\nabla f(x_{k(j)})\|^2) \\ &\leq \mathbb{E}(f(x_j) - f^*) - \frac{\gamma}{nL_{\max}R^2}\left(1 - \frac{\psi}{2}\gamma\right)\mathbb{E}((f(x_j) - f^*)^2) \\ &\leq \mathbb{E}(f(x_j) - f^*) - \frac{\gamma}{nL_{\max}R^2}\left(1 - \frac{\psi}{2}\gamma\right)(\mathbb{E}(f(x_j) - f^*))^2,\end{aligned}$$

where the third line uses the Jensen's inequality $\mathbb{E}(v^2) \geq (\mathbb{E}(v))^2$. Defining

$$C := \frac{\gamma}{nL_{\max}R^2}\left(1 - \frac{\psi}{2}\gamma\right),$$

we have

$$\begin{aligned}&\mathbb{E}(f(x_{j+1}) - f^*) \leq \mathbb{E}(f(x_j) - f^*) - C(\mathbb{E}(f(x_j) - f^*))^2 \\ \Rightarrow\ &\frac{1}{\mathbb{E}(f(x_j) - f^*)} \leq \frac{1}{\mathbb{E}(f(x_{j+1}) - f^*)} - C\frac{\mathbb{E}(f(x_j) - f^*)}{\mathbb{E}(f(x_{j+1}) - f^*)} \\ \Rightarrow\ &\frac{1}{\mathbb{E}(f(x_{j+1}) - f^*)} - \frac{1}{\mathbb{E}(f(x_j) - f^*)} \geq C\frac{\mathbb{E}(f(x_j) - f^*)}{\mathbb{E}(f(x_{j+1}) - f^*)} \geq C \\ \Rightarrow\ &\frac{1}{\mathbb{E}(f(x_{j+1}) - f^*)} \geq \frac{1}{f(x_0) - f^*} + C(j+1) \\ \Rightarrow\ &\mathbb{E}(f(x_{j+1}) - f^*) \leq \frac{1}{(f(x_0) - f^*)^{-1} + C(j+1)},\end{aligned}$$

which completes the proof of the sublinear rate (9). ■

Proof (Corollary 2) Note first that for ρ defined by (11), we have

$$\rho^\tau \leq \rho^{\tau+1} = \left(\left(1 + \frac{2eL_{\text{res}}}{\sqrt{n}L_{\max}}\right)^{\frac{\sqrt{n}L_{\max}}{2eL_{\text{res}}}}\right)^{\frac{2eL_{\text{res}}(\tau+1)}{\sqrt{n}L_{\max}}} \leq e^{\frac{2eL_{\text{res}}(\tau+1)}{\sqrt{n}L_{\max}}} \leq e,$$

and thus from the definition of ψ (5) that

$$\psi = 1 + \frac{2\tau\rho^\tau L_{\text{res}}}{\sqrt{n}L_{\max}} \leq 1 + \frac{2\tau e L_{\text{res}}}{\sqrt{n}L_{\max}} \leq 2. \tag{38}$$

We show now that the steplength parameter choice $\gamma = 1/\psi$ satisfies all the bounds in (6), by showing that the second and third bounds are implied by the first. For the second bound (6b), we have

$$\frac{(\rho-1)\sqrt{n}L_{\max}}{2\rho^{\tau+1}L_{\text{res}}} \geq \frac{(\rho-1)\sqrt{n}L_{\max}}{2eL_{\text{res}}} \geq 1 \geq \frac{1}{\psi},$$

where the second inequality follows from (11). For the third bound (6c), we have

$$\frac{(\rho-1)\sqrt{n}L_{\max}}{L_{\text{res}}\rho^\tau(2 + \frac{L_{\text{res}}}{\sqrt{n}L_{\max}})} = \frac{2eL_{\text{res}}}{L_{\text{res}}\rho^\tau(2 + \frac{L_{\text{res}}}{\sqrt{n}L_{\max}})} \geq \frac{2eL_{\text{res}}}{L_{\text{res}}e(2 + \frac{L_{\text{res}}}{\sqrt{n}L_{\max}})} = \frac{2}{2 + \frac{L_{\text{res}}}{\sqrt{n}L_{\max}}} \geq \frac{1}{\psi}.$$

We can thus set $\gamma = 1/\psi$, and by substituting this choice into (8) and using (38), we obtain (12). We obtain (13) by making the same substitution into (9). ■

Proof (Theorem 3) From Markov's inequality, we have

$$\begin{aligned}
\mathbb{P}(f(x_j) - f^* \geq \epsilon) &\leq \epsilon^{-1}\mathbb{E}(f(x_j) - f^*) \\
&\leq \epsilon^{-1}\left(1 - \frac{l}{2nL_{\max}}\right)^j (f(x_0) - f^*) \\
&\leq \epsilon^{-1}(1-c)^{(1/c)\left|\log \frac{f(x_0)-f^*}{\eta\epsilon}\right|}(f(x_0) - f^*) \quad \text{with } c = l/(2nL_{\max}) \\
&\leq \epsilon^{-1}(f(x_0) - f^*)e^{-\left|\log \frac{f(x_0)-f^*}{\eta\epsilon}\right|} \\
&= \eta e^{\log \frac{(f(x_0)-f^*)}{\eta\epsilon}} e^{-\left|\log \frac{f(x_0)-f^*}{\eta\epsilon}\right|} \\
&\leq \eta,
\end{aligned}$$

where the second inequality applies (12), the third inequality uses the definition of j (15), and the second last inequality uses the inequality $(1-c)^{1/c} \leq e^{-1}\ \forall\, c \in (0,1)$, which proves the essentially strongly convex case. Similarly, the general convex case is proven by

$$\mathbb{P}(f(x_j) - f^* \geq \epsilon) \leq \epsilon^{-1}\mathbb{E}(f(x_j) - f^*) \leq \frac{f(x_0) - f^*}{\epsilon\left(1 + j\frac{f(x_0)-f^*}{4nL_{\max}R^2}\right)} \leq \eta,$$

where the second inequality uses (13) and the last inequality uses the definition of j (16). ■

Appendix B. Proofs for Constrained Case

We start by introducing notation and proving several preliminary results. Define

$$(\Delta_j)_{i(j)} := (x_j - x_{j+1})_{i(j)}, \tag{39}$$

and formulate the update in Step 4 of Algorithm 1 in the following way:

$$x_{j+1} = \arg\min_{x\in\Omega} \langle \nabla_{i(j)} f(x_{k(j)}), (x - x_j)_{i(j)} \rangle + \frac{L_{\max}}{2\gamma}\|x - x_j\|^2.$$

(Note that $(x_{j+1})_i = (x_j)_i$ for $i \neq i(j)$.) From the optimality condition for this formulation, we have

$$\left\langle (x - x_{j+1})_{i(j)}, \nabla_{i(j)} f(x_{k(j)}) - \frac{L_{\max}}{\gamma}(\Delta_j)_{i(j)} \right\rangle \geq 0, \quad \text{for all } x \in \Omega.$$

This implies in particular that for all $x \in \Omega$, we have

$$\left\langle (\mathcal{P}_S(x) - x_{j+1})_{i(j)}, \nabla_{i(j)} f(x_{k(j)}) \right\rangle \geq \frac{L_{\max}}{\gamma} \left\langle (\mathcal{P}_S(x) - x_{j+1})_{i(j)}, (\Delta_j)_{i(j)} \right\rangle. \tag{40}$$

From the definition of $L_{\max}$, and using the notation (39), we have

$$f(x_{j+1}) \leq f(x_j) + \langle \nabla_{i(j)} f(x_j), -(\Delta_j)_{i(j)} \rangle + \frac{L_{\max}}{2}\|(\Delta_j)_{i(j)}\|^2,$$

which indicates that

$$\langle \nabla_{i(j)} f(x_j), (\Delta_j)_{i(j)} \rangle \leq f(x_j) - f(x_{j+1}) + \frac{L_{\max}}{2}\|(\Delta_j)_{i(j)}\|^2. \tag{41}$$

From optimality conditions for this definition, we have

$$\left\langle x - \bar{x}_{j+1}, \nabla f(x_{k(j)}) + \frac{L_{\max}}{\gamma}(\bar{x}_{j+1} - x_j) \right\rangle \geq 0 \quad \forall\, x \in \Omega. \tag{42}$$

We now define $\Delta_j := x_j - \bar{x}_{j+1}$, and note that this definition is consistent with $(\Delta)_{i(j)}$ defined in (39). It can be seen that

$$\mathbb{E}_{i(j)}(\|x_{j+1} - x_j\|^2) = \frac{1}{n}\|\bar{x}_{j+1} - x_j\|^2.$$

We now proceed to prove the main results of Section 5.

Proof (Theorem 4) We prove (20) by induction. First, note that for any vectors a and b, we have

$$\|a\|^2 - \|b\|^2 = 2\|a\|^2 - (\|a\|^2 + \|b\|^2) \leq 2\|a\|^2 - 2\langle a, b\rangle \leq 2\langle a, a - b\rangle \leq 2\|a\|\|a - b\|,$$

Thus for all j, we have

$$\|x_{j-1} - \bar{x}_j\|^2 - \|x_j - \bar{x}_{j+1}\|^2 \leq 2\|x_{j-1} - \bar{x}_j\|\|x_j - \bar{x}_{j+1} - x_{j-1} + \bar{x}_j\|. \tag{43}$$

The second factor in the r.h.s. of (43) is bounded as follows:

$$\begin{aligned}
&\|x_j - \bar{x}_{j+1} - x_{j-1} + \bar{x}_j\| \\
&\quad = \left\| x_j - \mathcal{P}_\Omega(x_j - \frac{\gamma}{L_{\max}}\nabla f(x_{k(j)})) - (x_{j-1} - \mathcal{P}_\Omega(x_{j-1} - \frac{\gamma}{L_{\max}}\nabla f(x_{k(j-1)}))) \right\| \\
&\quad \le \left\| x_j - \frac{\gamma}{L_{\max}}\nabla f(x_{k(j)}) - \mathcal{P}_\Omega(x_j - \frac{\gamma}{L_{\max}}\nabla f(x_{k(j)})) - (x_{j-1} - \frac{\gamma}{L_{\max}}\nabla f(x_{k(j-1)}) \right. \\
&\qquad \left. -\mathcal{P}_\Omega(x_{j-1} - \frac{\gamma}{L_{\max}}\nabla f(x_{k(j-1)}))) \right\| + \frac{\gamma}{L_{\max}} \left\| \nabla f(x_{k(j-1)}) - \nabla f(x_{k(j)}) \right\| \\
&\quad \le \left\| x_j - \frac{\gamma}{L_{\max}}\nabla f(x_{k(j)}) - x_{j-1} + \frac{\gamma}{L_{\max}}\nabla f(x_{k(j-1)}) \right\| \\
&\qquad + \frac{\gamma}{L_{\max}} \left\| \nabla f(x_{k(j-1)}) - \nabla f(x_{k(j)}) \right\| \\
&\quad \le \|x_j - x_{j-1}\| + 2\frac{\gamma}{L_{\max}} \left\| \nabla f(x_{k(j)}) - \nabla f(x_{k(j-1)}) \right\| \\
&\quad \le \|x_j - x_{j-1}\| + 2\frac{\gamma}{L_{\max}} \sum_{d=\min\{k(j-1),k(j)\}}^{\max\{k(j-1),k(j)\}-1} \|\nabla f(x_d) - \nabla f(x_{d+1})\| \\
&\quad \le \|x_j - x_{j-1}\| + 2\frac{\gamma L_{\text{res}}}{L_{\max}} \sum_{d=\min\{k(j-1),k(j)\}}^{\max\{k(j-1),k(j)\}-1} \|x_d - x_{d+1}\|, \qquad (44)
\end{aligned}$$

where the first inequality follows by adding and subtracting a term, and the second inequality uses the nonexpansive property of projection:

$$\|(z - \mathcal{P}_\Omega(z)) - (y - \mathcal{P}_\Omega(y))\| \le \|z - y\|.$$

One can see that $j - 1 - \tau \le k(j-1) \le j - 1$ and $j - \tau \le k(j) \le j$, which implies that $j - 1 - \tau \le d \le j - 1$ for each index d in the summation in (44). It also follows that

$$\max\{k(j-1), k(j)\} - 1 - \min\{k(j-1), k(j)\} \le \tau. \qquad (45)$$

We set $j = 1$, and note that $k(0) = 0$ and $k(1) \le 1$. Thus, in this case, we have that the lower and upper limits of the summation in (44) are 0 and 0, respectively. Thus, this summation is vacuous, and we have

$$\|x_1 - \bar{x}_2 + x_0 - \bar{x}_1\| \le \left(1 + 2\frac{\gamma L_{\text{res}}}{L_{\max}}\right) \|x_1 - x_0\|,$$

By substituting this bound in (43) and setting $j = 1$, we obtain

$$\mathbb{E}(\|x_0 - \bar{x}_1\|^2) - \mathbb{E}(\|x_1 - \bar{x}_2\|^2) \le \left(2 + 4\frac{\gamma L_{\text{res}}}{L_{\max}}\right) \mathbb{E}(\|x_1 - x_0\| \|\bar{x}_1 - x_0\|). \qquad (46)$$

For any j, we have

$$\begin{aligned}\mathbb{E}(\|x_j - x_{j-1}\|\|\bar{x}_j - x_{j-1}\|) &\leq \frac{1}{2}\mathbb{E}(n^{1/2}\|x_j - x_{j-1}\|^2 + n^{-1/2}\|\bar{x}_j - x_{j-1}\|^2) \\ &= \frac{1}{2}\mathbb{E}(n^{1/2}\mathbb{E}_{i(j-1)}(\|x_j - x_{j-1}\|^2) + n^{-1/2}\|\bar{x}_j - x_{j-1}\|^2) \\ &= \frac{1}{2}\mathbb{E}(n^{-1/2}\|\bar{x}_j - x_{j-1}\|^2 + n^{-1/2}\|\bar{x}_j - x_{j-1}\|^2) \\ &= n^{-1/2}\mathbb{E}\|\bar{x}_j - x_{j-1}\|^2. \end{aligned} \tag{47}$$

Returning to (46), we have

$$\mathbb{E}(\|x_0 - \bar{x}_1\|^2) - \mathbb{E}(\|x_1 - \bar{x}_2\|^2) \leq 2n^{-1/2}\mathbb{E}\|\bar{x}_1 - x_0\|^2$$

which implies that

$$\mathbb{E}(\|x_0 - \bar{x}_1\|^2) \leq \left(1 - \frac{2}{\sqrt{n}} - \frac{4\gamma L_{\rm res}}{\sqrt{n}L_{\max}}\right)^{-1} \mathbb{E}(\|x_1 - \bar{x}_2\|^2) \leq \rho\mathbb{E}(\|x_1 - \bar{x}_2\|^2).$$

To see the last inequality above, we only need to verify that

$$\gamma \leq \left(1 - \rho^{-1} - \frac{2}{\sqrt{n}}\right)\frac{\sqrt{n}L_{\max}}{4L_{\rm res}}.$$

This proves that (20) holds for $j = 1$.

To take the inductive step, we assume that show that (20) holds up to index $j-1$. We have for $j - 1 - \tau \leq d \leq j - 2$ that

$$\begin{aligned}\mathbb{E}(\|x_d - x_{d+1}\|\|\bar{x}_j - x_{j-1}\|) &\leq \frac{1}{2}\mathbb{E}(n^{1/2}\|x_d - x_{d+1}\|^2 + n^{-1/2}\|\bar{x}_j - x_{j-1}\|^2) \\ &= \frac{1}{2}\mathbb{E}(n^{1/2}\mathbb{E}_{i(d)}(\|x_d - x_{d+1}\|^2) + n^{-1/2}\|\bar{x}_j - x_{j-1}\|^2) \\ &= \frac{1}{2}\mathbb{E}(n^{-1/2}\|x_d - \bar{x}_{d+1}\|^2 + n^{-1/2}\|\bar{x}_j - x_{j-1}\|^2) \\ &\leq \frac{1}{2}\mathbb{E}(n^{-1/2}\rho^{\tau}\|x_{j-1} - \bar{x}_j\|^2 + n^{-1/2}\|\bar{x}_j - x_{j-1}\|^2) \\ &\leq \frac{\rho^{\tau}}{n^{1/2}}\mathbb{E}(\|\bar{x}_j - x_{j-1}\|^2), \end{aligned} \tag{48}$$

where the second inequality uses the inductive hypothesis. By substituting (44) into (43) and taking expectation on both sides of (43), we obtain

$$\begin{aligned}
&\mathbb{E}(\|x_{j-1}-\bar{x}_j\|^2)-\mathbb{E}(\|x_j-\bar{x}_{j+1}\|^2)\\
&\quad\le 2\mathbb{E}(\|\bar{x}_j-x_{j-1}\|\|\bar{x}_j-\bar{x}_{j+1}+x_j-x_{j-1}\|)\\
&\quad\le 2\mathbb{E}\left(\|\bar{x}_j-x_{j-1}\|\left(\|x_j-x_{j-1}\|+2\frac{\gamma L_{\text{res}}}{L_{\max}}\sum_{d=\min\{k(j-1),k(j)\}}^{\max\{k(j-1),k(j)\}-1}\|x_d-x_{d+1}\|\right)\right)\\
&\quad= 2\mathbb{E}(\|\bar{x}_j-x_{j-1}\|\|x_j-x_{j-1}\|)+\\
&\qquad\qquad 4\frac{\gamma L_{\text{res}}}{L_{\max}}\sum_{d=\min\{k(j-1),k(j)\}}^{\max\{k(j-1),k(j)\}-1}\mathbb{E}(\|\bar{x}_j-x_{j-1}\|\|x_d-x_{d+1}\|)\\
&\quad\le n^{-1/2}\left(2+\frac{4\gamma L_{\text{res}}\tau\rho^\tau}{L_{\max}}\right)\mathbb{E}(\|x_{j-1}-\bar{x}_j\|^2),
\end{aligned}$$

where the last line uses (45), (47), and (48). It follows that

$$\mathbb{E}(\|x_{j-1}-\bar{x}_j\|^2)\le\left(1-n^{-1/2}\left(2+\frac{4\gamma L_{\text{res}}\tau\rho^\tau}{L_{\max}}\right)\right)^{-1}\mathbb{E}(\|x_j-\bar{x}_{j+1}\|^2)\le\rho\mathbb{E}(\|x_j-\bar{x}_{j+1}\|^2).$$

To see the last inequality, one only needs to verify that

$$\rho^{-1}\le 1-\frac{1}{\sqrt{n}}\left(2+\frac{4\gamma L_{\text{res}}\tau\rho^\tau}{L_{\max}}\right)\;\Leftrightarrow\;\gamma\le\left(1-\rho^{-1}-\frac{2}{\sqrt{n}}\right)\frac{\sqrt{n}L_{\max}}{4L_{\text{res}}\tau\rho^\tau},$$

and the last inequality is true because of the upper bound of γ in (19). It proves (20).

Next we will show the expectation of objective is monotonically decreasing. We have using the definition (39) that

$$\begin{aligned}
&\mathbb{E}_{i(j)}(f(x_{j+1}))=n^{-1}\sum_{i=1}^n f(x_j+(\Delta_j)_i)\\
&\quad\le n^{-1}\sum_{i=1}^n\left[f(x_j)+\langle\nabla_i f(x_j),(\bar{x}_{j+1}-x_j)_i\rangle+\frac{L_{\max}}{2}\|(x_{j+1}-x_j)_i\|^2\right]\\
&\quad= f(x_j)+n^{-1}\left(\langle\nabla f(x_j),\bar{x}_{j+1}-x_j\rangle+\frac{L_{\max}}{2}\|\bar{x}_{j+1}-x_j\|^2\right)\\
&\quad= f(x_j)+\frac{1}{n}\left(\langle\nabla f(x_{k(j)}),\bar{x}_{j+1}-x_j\rangle+\frac{L_{\max}}{2}\|\bar{x}_{j+1}-x_j\|^2\right)+\frac{1}{n}\langle\nabla f(x_j)-\nabla f(x_{k(j)}),\bar{x}_{j+1}-x_j\rangle\\
&\quad\le f(x_j)+\frac{1}{n}\left(\frac{L_{\max}}{2}\|\bar{x}_{j+1}-x_j\|^2-\frac{L_{\max}}{\gamma}\|\bar{x}_{j+1}-x_j\|^2\right)+\frac{1}{n}\langle\nabla f(x_j)-\nabla f(x_{k(j)}),\bar{x}_{j+1}-x_j\rangle\\
&\quad= f(x_j)-\left(\frac{1}{\gamma}-\frac{1}{2}\right)\frac{L_{\max}}{n}\|\bar{x}_{j+1}-x_j\|^2+\frac{1}{n}\langle\nabla f(x_j)-\nabla f(x_{k(j)}),\bar{x}_{j+1}-x_j\rangle,
\end{aligned}\tag{49}$$

where the second inequality uses (42). Consider the expectation of the last term on the right-hand side of this expression. We have

$$\begin{aligned}
&\mathbb{E}\langle \nabla f(x_j) - \nabla f(x_{k(j)}), \bar{x}_{j+1} - x_j \rangle \\
&\quad \le \mathbb{E}\|\nabla f(x_j) - \nabla f(x_{k(j)})\| \|\bar{x}_{j+1} - x_j\| \\
&\quad \le \mathbb{E} \sum_{d=k(j)}^{j-1} \|\nabla f(x_d) - \nabla f(x_{d+1})\| \|\bar{x}_{j+1} - x_j\| \\
&\quad \le L_{\text{res}} \mathbb{E} \sum_{d=k(j)}^{j-1} \|x_d - x_{d+1}\| \|\bar{x}_{j+1} - x_j\| \\
&\quad \le \frac{L_{\text{res}}}{2} \mathbb{E} \sum_{d=k(j)}^{j-1} (n^{1/2}\|x_d - x_{d+1}\|^2 + n^{-1/2}\|\bar{x}_{j+1} - x_j\|^2) \\
&\quad = \frac{L_{\text{res}}}{2} \mathbb{E} \sum_{d=k(j)}^{j-1} (n^{1/2}\mathbb{E}_{i(d)}(\|x_d - x_{d+1}\|^2) + n^{-1/2}\|\bar{x}_{j+1} - x_j\|^2) \\
&\quad = \frac{L_{\text{res}}}{2} \mathbb{E} \sum_{d=k(j)}^{j-1} (n^{-1/2}\|x_d - \bar{x}_{d+1}\|^2 + n^{-1/2}\|\bar{x}_{j+1} - x_j\|^2) \\
&\quad \le \frac{L_{\text{res}}}{2n^{1/2}} \mathbb{E} \sum_{d=k(j)}^{j-1} (1+\rho^\tau)\|\bar{x}_{j+1} - x_j\|^2 \\
&\quad \le \frac{L_{\text{res}}\tau\rho^\tau}{n^{1/2}} \mathbb{E}\|\bar{x}_{j+1} - x_j\|^2, \qquad (50)
\end{aligned}$$

where the fifth inequality uses (20). By taking expectation on both sides of (49) and substituting (50), we have

$$\mathbb{E}(f(x_{j+1})) \le \mathbb{E}(f(x_j)) - \frac{1}{n}\left(\left(\frac{1}{\gamma} - \frac{1}{2}\right) L_{\max} - \frac{L_{\text{res}}\tau\rho^\tau}{n^{1/2}}\right) \mathbb{E}\|\bar{x}_{j+1} - x_j\|^2.$$

To see $\left(\frac{1}{\gamma} - \frac{1}{2}\right) L_{\max} - \frac{L_{\text{res}}\tau\rho^\tau}{n^{1/2}} \ge 0$, we only need to verify

$$\gamma \le \left(\frac{1}{2} + \frac{L_{\text{res}}\tau\rho^\tau}{\sqrt{n}L_{\max}}\right)^{-1}$$

which is implied by the first upper bound of γ (19). Therefore, we have proved the monotonicity $\mathbb{E}(f(x_{j+1})) \le \mathbb{E}(f(x_j))$.

Next we prove the sublinear convergence rate for the constrained smooth convex case in (22). We have

$$
\begin{aligned}
&\|x_{j+1} - \mathcal{P}_S(x_{j+1})\|^2 \leq \|x_{j+1} - \mathcal{P}_S(x_j)\|^2 \\
&\quad = \|x_j - (\Delta_j)_{i(j)} e_{i(j)} - \mathcal{P}_S(x_j)\|^2 \\
&\quad = \|x_j - \mathcal{P}_S(x_j)\|^2 + |(\Delta_j)_{i(j)}|^2 - 2(x_j - \mathcal{P}_S(x_j))_{i(j)}(\Delta_j)_{i(j)} \\
&\quad = \|x_j - \mathcal{P}_S(x_j)\|^2 - |(\Delta_j)_{i(j)}|^2 - 2\left((x_j - \mathcal{P}_S(x_j))_{i(j)} - (\Delta_j)_{i(j)}\right)(\Delta_j)_{i(j)} \\
&\quad = \|x_j - \mathcal{P}_S(x_j)\|^2 - \|(\Delta_j)_{i(j)}\|^2 + 2(\mathcal{P}_S(x_j) - x_{j+1})_{i(j)}(\Delta_j)_{i(j)} \\
&\quad \leq \|x_j - \mathcal{P}_S(x_j)\|^2 - |(\Delta_j)_{i(j)}|^2 + \frac{2\gamma}{L_{\max}}(\mathcal{P}_S(x_j) - x_{j+1})_{i(j)} \nabla_{i(j)} f(x_{k(j)}) \\
&\quad = \|x_j - \mathcal{P}_S(x_j)\|^2 - |(\Delta_j)_{i(j)}|^2 + \frac{2\gamma}{L_{\max}}(\mathcal{P}_S(x_j) - x_j)_{i(j)} \nabla_{i(j)} f(x_{k(j)}) + \\
&\qquad \frac{2\gamma}{L_{\max}}\left((\Delta_j)_{i(j)} \nabla_{i(j)} f(x_j) + (\Delta_j)_{i(j)}\left(\nabla_{i(j)} f(x_{k(j)}) - \nabla_{i(j)} f(x_j)\right)\right) \\
&\quad \leq \|x_j - \mathcal{P}_S(x_j)\|^2 - |(\Delta_j)_{i(j)}|^2 + \frac{2\gamma}{L_{\max}}(\mathcal{P}_S(x_j) - x_j)_{i(j)} \nabla_{i(j)} f(x_{k(j)}) + \\
&\qquad \frac{2\gamma}{L_{\max}}\Bigg(f(x_j) - f(x_{j+1}) + \frac{L_{\max}}{2}|(\Delta_j)_{i(j)}|^2 \\
&\qquad + (\Delta_j)_{i(j)}\left(\nabla_{i(j)} f(x_{k(j)}) - \nabla_{i(j)} f(x_j)\right)\Bigg) \\
&\quad = \|x_j - \mathcal{P}_S(x_j)\|^2 - (1-\gamma)|(\Delta_j)_{i(j)}|^2 + \frac{2\gamma}{L_{\max}} \underbrace{(\mathcal{P}_S(x_j) - x_j)_{i(j)} \nabla_{i(j)} f(x_{k(j)})}_{T_1} + \\
&\qquad \frac{2\gamma}{L_{\max}}(f(x_j) - f(x_{j+1})) + \frac{2\gamma}{L_{\max}} \underbrace{(\Delta_j)_{i(j)}\left(\nabla_{i(j)} f(x_{k(j)}) - \nabla_{i(j)} f(x_j)\right)}_{T_2}, \qquad (51)
\end{aligned}
$$

where the second inequality uses (40) and the third inequality uses (41). We now seek upper bounds on the quantities T_1 and T_2 in the expectation sense. For T_1, we have

$$\begin{aligned}
\mathbb{E}(T_1) &= n^{-1}\mathbb{E}\langle \mathcal{P}_S(x_j) - x_j, \nabla f(x_{k(j)})\rangle \\
&= n^{-1}\mathbb{E}\langle \mathcal{P}_S(x_j) - x_{k(j)}, \nabla f(x_{k(j)})\rangle + n^{-1}\mathbb{E}\sum_{d=k(j)}^{j-1}\langle x_d - x_{d+1}, \nabla f(x_{k(j)})\rangle \\
&= n^{-1}\mathbb{E}\langle \mathcal{P}_S(x_j) - x_{k(j)}, \nabla f(x_{k(j)})\rangle \\
&\quad + n^{-1}\mathbb{E}\sum_{d=k(j)}^{j-1}\langle x_d - x_{d+1}, \nabla f(x_d)\rangle + \langle x_d - x_{d+1}, \nabla f(x_{k(j)}) - \nabla f(x_d)\rangle \\
&\leq n^{-1}\mathbb{E}(f^* - f(x_{k(j)})) + n^{-1}\mathbb{E}\sum_{d=k(j)}^{j-1}\left(f(x_d) - f(x_{d+1}) + \frac{L_{\max}}{2}\|x_d - x_{d+1}\|^2\right) \\
&\quad + n^{-1}\mathbb{E}\sum_{d=k(j)}^{j-1}\langle x_d - x_{d+1}, \nabla f(x_{k(j)}) - \nabla f(x_d)\rangle \\
&= n^{-1}\mathbb{E}(f^* - f(x_j)) + \frac{L_{\max}}{2n}\mathbb{E}\sum_{d=k(j)}^{j-1}\|x_d - x_{d+1}\|^2 \\
&\quad + n^{-1}\mathbb{E}\sum_{d=k(j)}^{j-1}\langle x_d - x_{d+1}, \nabla f(x_{k(j)}) - \nabla f(x_d)\rangle \\
&= n^{-1}\mathbb{E}(f^* - f(x_j)) + \frac{L_{\max}}{2n^2}\mathbb{E}\sum_{d=k(j)}^{j-1}\|x_d - \bar{x}_{d+1}\|^2 \\
&\quad + n^{-1}\mathbb{E}\sum_{d=k(j)}^{j-1}\langle x_d - x_{d+1}, \nabla f(x_{k(j)}) - \nabla f(x_d)\rangle \\
&\leq n^{-1}\mathbb{E}(f^* - f(x_j)) + \frac{L_{\max}\tau\rho^\tau}{2n^2}\mathbb{E}\|x_j - \bar{x}_{j+1}\|^2 \\
&\quad + n^{-1}\sum_{d=k(j)}^{j-1}\underbrace{\mathbb{E}\langle x_d - x_{d+1}, \nabla f(x_{k(j)}) - \nabla f(x_d)\rangle}_{T_3},
\end{aligned}$$

where the last inequality uses (20). The upper bound of $\mathbb{E}(T_3)$ is estimated by

$$\begin{aligned}
\mathbb{E}(T_3) &= \mathbb{E}\langle x_d - x_{d+1}, \nabla f(x_{k(j)}) - \nabla f(x_d)\rangle \\
&= \mathbb{E}(\mathbb{E}_{i(d)}\langle x_d - x_{d+1}, \nabla f(x_{k(j)}) - \nabla f(x_d)\rangle) \\
&= n^{-1}\mathbb{E}\langle x_d - \bar{x}_{d+1}, \nabla f(x_{k(j)}) - \nabla f(x_d)\rangle \\
&\leq n^{-1}\mathbb{E}\|x_d - \bar{x}_{d+1}\|\|\nabla f(x_{k(j)}) - \nabla f(x_d)\| \\
&\leq n^{-1}\mathbb{E}(\|x_d - \bar{x}_{d+1}\| \sum_{t=k(j)}^{d-1} \|\nabla f(x_t) - \nabla f(x_{t+1})\|) \\
&\leq \frac{L_{\mathrm{res}}}{n} \sum_{t=k(j)}^{d-1} \mathbb{E}(\|x_d - \bar{x}_{d+1}\|\|x_t - x_{t+1}\|) \\
&\leq \frac{L_{\mathrm{res}}}{2n} \sum_{t=k(j)}^{d-1} \mathbb{E}(n^{-1/2}\|x_d - \bar{x}_{d+1}\|^2 + n^{1/2}\|x_t - x_{t+1}\|^2) \\
&\leq \frac{L_{\mathrm{res}}}{2n} \sum_{t=k(j)}^{d-1} \mathbb{E}(n^{-1/2}\|x_d - \bar{x}_{d+1}\|^2 + n^{-1/2}\|x_t - \bar{x}_{t+1}\|^2) \\
&\leq \frac{L_{\mathrm{res}}\rho^\tau}{n^{3/2}} \sum_{t=k(j)}^{d-1} \mathbb{E}(\|x_j - \bar{x}_{j+1}\|^2) \\
&\leq \frac{L_{\mathrm{res}}\tau\rho^\tau}{n^{3/2}}\mathbb{E}(\|x_j - \bar{x}_{j+1}\|^2).
\end{aligned}$$

where the second last inequality uses (20). Therefore, $\mathbb{E}(T_1)$ can be bounded by

$$\begin{aligned}
\mathbb{E}(T_1) &= \mathbb{E}\langle(\mathcal{P}_S(x_j) - x_j)_{i(j)}, \nabla_{i(j)} f(x_{k(j)})\rangle \\
&\leq \frac{1}{n}\mathbb{E}(f^* - f(x_j)) + \frac{L_{\max}\tau\rho^\tau}{2n^2}\mathbb{E}\|x_j - \bar{x}_{j+1}\|^2 + \sum_{d=k(j)}^{j-1} \frac{L_{\mathrm{res}}\tau\rho^\tau}{n^{5/2}}\mathbb{E}(\|x_j - \bar{x}_{j+1}\|^2) \\
&= \frac{1}{n}\left(f^* - \mathbb{E}f(x_j) + \left(\frac{L_{\max}\tau\rho^\tau}{2n} + \frac{L_{\mathrm{res}}\tau^2\rho^\tau}{n^{3/2}}\right)\mathbb{E}(\|x_j - \bar{x}_{j+1}\|^2)\right). \qquad (52)
\end{aligned}$$

For T_2, we have

$$\begin{aligned}
\mathbb{E}(T_2) &= \mathbb{E}(\Delta_j)_{i(j)} \left(\nabla_{i(j)} f(x_{k(j)}) - \nabla_{i(j)} f(x_j)\right) \\
&= n^{-1}\mathbb{E}\langle \Delta_j, \nabla f(x_{k(j)}) - \nabla f(x_j)\rangle \\
&\leq n^{-1}\mathbb{E}(\|\Delta_j\|\|\nabla f(x_{k(j)}) - \nabla f(x_j)\|) \\
&\leq n^{-1}\mathbb{E}\left(\sum_{d=k(j)}^{j-1} \|\Delta_j\|\|\nabla f(x_d) - \nabla f(x_{d+1})\|\right) \\
&\leq \frac{L_{\text{res}}}{n}\mathbb{E}\left(\sum_{d=k(j)}^{j-1} \|\Delta_j\|\|x_d - x_{d+1}\|\right) \\
&= \frac{L_{\text{res}}}{2n}\mathbb{E}\left(\sum_{d=k(j)}^{j-1} n^{-1/2}\|\Delta_j\|^2 + n^{1/2}\|x_d - x_{d+1}\|^2\right) \\
&= \frac{L_{\text{res}}}{2n}\mathbb{E}\left(\sum_{d=k(j)}^{j-1} n^{-1/2}\|x_j - \bar{x}_{j+1}\|^2 + n^{1/2}\mathbb{E}_{i(d)}\|x_d - x_{d+1}\|^2\right) \\
&= \frac{L_{\text{res}}}{2n}\mathbb{E}\left(\sum_{d=k(j)}^{j-1} n^{-1/2}\|x_j - \bar{x}_{j+1}\|^2 + n^{-1/2}\|x_d - \bar{x}_{d+1}\|^2\right) \\
&= \frac{L_{\text{res}}}{2n^{3/2}}\left(\sum_{d=k(j)}^{j-1} \mathbb{E}\|x_j - \bar{x}_{j+1}\|^2 + \mathbb{E}\|x_d - \bar{x}_{d+1}\|^2\right) \\
&\leq \frac{L_{\text{res}}(1+\rho^\tau)}{2n^{3/2}} \sum_{d=k(j)}^{j-1} \mathbb{E}\|x_j - \bar{x}_{j+1}\|^2 \\
&\leq \frac{L_{\text{res}}\tau\rho^\tau}{n^{3/2}}\mathbb{E}\|x_j - \bar{x}_{j+1}\|^2, \qquad (53)
\end{aligned}$$

where the second last inequality uses (20).

By taking the expectation on both sides of (51), using $\mathbb{E}_{i(j)}(|(\Delta_j)_{i(j)}|^2) = n^{-1}\|x_j - \bar{x}_{j+1}\|^2$, and substituting the upper bounds from (52) and (53), we obtain

$$\begin{aligned}
\mathbb{E}\|x_{j+1} - \mathcal{P}_S(x_{j+1})\|^2 &\leq \mathbb{E}\|x_j - \mathcal{P}_S(x_j)\|^2 \\
&\quad - \frac{1}{n}\left(1 - \gamma - \frac{2\gamma L_{\text{res}}\tau\rho^\tau}{L_{\max}n^{1/2}} - \frac{\gamma\tau\rho^\tau}{n} - \frac{2\gamma L_{\text{res}}\tau^2\rho^\tau}{L_{\max}n^{3/2}}\right)\mathbb{E}\|x_j - \bar{x}_{j+1}\|^2 \\
&\quad + \frac{2\gamma}{L_{\max}n}(f^* - \mathbb{E}f(x_j)) + \frac{2\gamma}{L_{\max}}(\mathbb{E}f(x_j) - \mathbb{E}f(x_{j+1})) \\
&\leq \mathbb{E}\|x_j - \mathcal{P}_S(x_j)\|^2 + \frac{2\gamma}{L_{\max}n}(f^* - \mathbb{E}f(x_j)) + \frac{2\gamma}{L_{\max}}(\mathbb{E}f(x_j) - \mathbb{E}f(x_{j+1})). \qquad (54)
\end{aligned}$$

In the second inequality, we were able to drop the term involving $\mathbb{E}\|x_j - \bar{x}_{j+1}\|^2$ by using the fact that

$$1 - \gamma - \frac{2\gamma L_{\text{res}}\tau\rho^\tau}{L_{\max}n^{1/2}} - \frac{\gamma\tau\rho^\tau}{n} - \frac{2\gamma L_{\text{res}}\tau^2\rho^\tau}{L_{\max}n^{3/2}} = 1 - \gamma\psi \geq 0,$$

which follows from the definition (18) of ψ and from the first upper bound on γ in (19). It follows that

$$\begin{aligned}
&\mathbb{E}\|x_{j+1} - \mathcal{P}_S(x_{j+1})\|^2 + \frac{2\gamma}{L_{\max}}(\mathbb{E}f(x_{j+1}) - f^*) \\
&\quad \leq \mathbb{E}\|x_j - \mathcal{P}_S(x_j)\|^2 + \frac{2\gamma}{L_{\max}}(\mathbb{E}f(x_j) - f^*) - \frac{2\gamma}{L_{\max} n}(\mathbb{E}f(x_j) - f^*) \qquad (55)\\
&\quad \leq \|x_0 - \mathcal{P}_S(x_0)\|^2 + \frac{2\gamma}{L_{\max}}(f(x_0) - f^*) - \frac{2\gamma}{L_{\max} n}\sum_{t=0}^{j}(\mathbb{E}f(x_t) - f^*) \\
&\quad \leq R^2 + \frac{2\gamma}{L_{\max}}(f(x_0) - f^*) - \frac{2\gamma(j+1)}{L_{\max} n}(\mathbb{E}f(x_{j+1}) - f^*),
\end{aligned}$$

where the second inequality follows by applying induction to the inequality

$$S_{j+1} \leq S_j - \frac{2\gamma}{L_{\max} n}\mathbb{E}(f(x_j) - f^*),$$

where

$$S_j := \mathbb{E}(\|x_j - \mathcal{P}_S(x_j)\|^2) + \frac{2\gamma}{L_{\max}}\mathbb{E}(f(x_j) - \mathcal{P}_S(x_j)),$$

and the last line uses the monotonicity of $\mathbb{E}f(x_j)$ (proved above) and the assumed bound $\|x_0 - \mathcal{P}_S(x_0)\| \leq R$. It implies that

$$\begin{aligned}
&\mathbb{E}\|x_{j+1} - \mathcal{P}_S(x_{j+1})\|^2 + \frac{2\gamma}{L_{\max}}(\mathbb{E}f(x_{j+1}) - f^*) + \frac{2\gamma(j+1)}{L_{\max} n}(\mathbb{E}f(x_{j+1}) - f^*) \\
&\quad \leq R^2 + \frac{2\gamma}{L_{\max}}(f(x_0) - f^*) \\
&\Rightarrow \frac{2\gamma(n+j+1)}{L_{\max} n}(\mathbb{E}f(x_{j+1}) - f^*) \leq R^2 + \frac{2\gamma}{L_{\max}}(f(x_0) - f^*) \\
&\Rightarrow \mathbb{E}f(x_{j+1}) - f^* \leq \frac{n(R^2 L_{\max} + 2\gamma(f(x_0) - f^*))}{2\gamma(n+j+1)}.
\end{aligned}$$

This completes the proof of the sublinear convergence rate (22).

Finally, we prove the linear convergence rate (21) for the essentially strongly convex case. All bounds proven above hold, and we make use the following additional property:

$$f(x_j) - f^* \geq \langle \nabla f(\mathcal{P}_S(x_j)), x_j - \mathcal{P}_S(x_j)\rangle + \frac{l}{2}\|x_j - \mathcal{P}_S(x_j)\|^2 \geq \frac{l}{2}\|x_j - \mathcal{P}_S(x_j)\|^2,$$

due to feasibility of x_j and $\langle \nabla f(\mathcal{P}_S(x_j)), x_j - \mathcal{P}_S(x_j)\rangle \geq 0$. By using this result together with some elementary manipulation, we obtain

$$\begin{aligned}
f(x_j) - f^* &= \left(1 - \frac{L_{\max}}{l\gamma + L_{\max}}\right)(f(x_j) - f^*) + \frac{L_{\max}}{l\gamma + L_{\max}}(f(x_j) - f^*) \\
&\geq \left(1 - \frac{L_{\max}}{l\gamma + L_{\max}}\right)(f(x_j) - f^*) + \frac{L_{\max} l}{2(l\gamma + L_{\max})}\|x_j - \mathcal{P}_S(x_j)\|^2 \\
&= \frac{L_{\max} l}{2(l\gamma + L_{\max})}\left(\|x_j - \mathcal{P}_S(x_j)\|^2 + \frac{2\gamma}{L_{\max}}(f(x_j) - f^*)\right). \qquad (56)
\end{aligned}$$

Recalling (55), we have

$$\begin{aligned}&\mathbb{E}\|x_{j+1}-\mathcal{P}_S(x_{j+1})\|^2+\frac{2\gamma}{L_{\max}}(\mathbb{E}f(x_{j+1})-f^*)\\&\quad\le \mathbb{E}\|x_j-\mathcal{P}_S(x_j)\|^2+\frac{2\gamma}{L_{\max}}(\mathbb{E}f(x_j)-f^*)-\frac{2\gamma}{L_{\max}n}(\mathbb{E}f(x_j)-f^*).\end{aligned}\tag{57}$$

By taking the expectation of both sides in (56) and substituting in the last term of (57), we obtain

$$\begin{aligned}&\mathbb{E}\|x_{j+1}-\mathcal{P}_S(x_{j+1})\|^2+\frac{2\gamma}{L_{\max}}(\mathbb{E}f(x_{j+1})-f^*)\\&\le \mathbb{E}\|x_j-\mathcal{P}_S(x_j)\|^2+\frac{2\gamma}{L_{\max}}(\mathbb{E}f(x_j)-f^*)\\&\quad-\frac{2\gamma}{L_{\max}n}\left(\frac{L_{\max}l}{2(l\gamma+L_{\max})}\left(\mathbb{E}\|x_j-\mathcal{P}_S(x_j)\|^2+\frac{2\gamma}{L_{\max}}(\mathbb{E}f(x_j)-f^*)\right)\right)\\&=\left(1-\frac{l}{n(l+\gamma^{-1}L_{\max})}\right)\left(\mathbb{E}\|x_j-\mathcal{P}_S(x_j)\|^2+\frac{2\gamma}{L_{\max}}(\mathbb{E}f(x_j)-f^*)\right)\\&\le\left(1-\frac{l}{n(l+\gamma^{-1}L_{\max})}\right)^{j+1}\left(\|x_0-\mathcal{P}_S(x_0)\|^2+\frac{2\gamma}{L_{\max}}(f(x_0)-f^*)\right),\end{aligned}$$

which yields (21). ■

Proof (Corollary 5) To apply Theorem 4, we first show $\rho>\left(1-\frac{2}{\sqrt{n}}\right)^{-1}$. Using the bound (23), together with $L_{\rm res}/L_{\max}\ge 1$, we obtain

$$\begin{aligned}&\left(1-\frac{2}{\sqrt{n}}\right)\left(1+\frac{4e\tau L_{\rm res}}{\sqrt{n}L_{\max}}\right)=\left(1+\frac{4e\tau L_{\rm res}}{\sqrt{n}L_{\max}}\right)-\left(1+\frac{4e\tau L_{\rm res}}{\sqrt{n}L_{\max}}\right)\frac{2}{\sqrt{n}}\\&\ge\left(1+\frac{4e\tau}{\sqrt{n}}\right)-\left(1+\frac{1}{\tau+1}\right)\frac{2}{\sqrt{n}}=1+\left(2e\tau-1-\frac{1}{\tau+1}\right)\frac{2}{\sqrt{n}}>1,\end{aligned}$$

where the last inequality uses $\tau\ge 1$. Note that for ρ defined by (24), and using (23), we have

$$\rho^\tau\le\rho^{\tau+1}=\left(\left(1+\frac{4e\tau L_{\rm res}}{\sqrt{n}L_{\max}}\right)^{\frac{\sqrt{n}L_{\max}}{4e\tau L_{\rm res}}}\right)^{\frac{4e\tau L_{\rm res}(\tau+1)}{\sqrt{n}L_{\max}}}\le e^{\frac{4e\tau L_{\rm res}(\tau+1)}{\sqrt{n}L_{\max}}}\le e.$$

Thus from the definition of ψ (18), we have that

$$\begin{aligned}\psi&=1+\frac{L_{\rm res}\tau\rho^\tau}{\sqrt{n}L_{\max}}\left(2+\frac{L_{\max}}{\sqrt{n}L_{\rm res}}+\frac{2\tau}{n}\right)\le 1+\frac{L_{\rm res}\tau\rho^\tau}{4eL_{\rm res}\tau(\tau+1)}\left(2+\frac{1}{\sqrt{n}}+\frac{2\tau}{n}\right)\\&\le 1+\frac{1}{4(\tau+1)}\left(2+\frac{1}{\sqrt{n}}+\frac{2\tau}{n}\right)\le 1+\left(\frac{1}{4}+\frac{1}{16}+\frac{1}{10}\right)\le 2.\end{aligned}\tag{58}$$

(The second last inequality uses $n\ge 5$ and $\tau\ge 1$.) Thus, the steplength parameter choice $\gamma=1/2$ satisfies the first bound in (19). To show that the second bound in (19) holds also,

we have

$$\left(1 - \frac{1}{\rho} - \frac{2}{\sqrt{n}}\right) \frac{\sqrt{n} L_{\max}}{4 L_{\text{res}} \tau \rho^{\tau}} = \left(\frac{\rho - 1}{\rho} - \frac{2}{\sqrt{n}}\right) \frac{\sqrt{n} L_{\max}}{4 L_{\text{res}} \tau \rho^{\tau}}$$
$$= \frac{4 e \tau L_{\text{res}}}{4 L_{\text{res}} \tau \rho^{\tau+1}} - \frac{L_{\max}}{2 L_{\text{res}} \tau \rho^{\tau}} \geq 1 - \frac{1}{2} = \frac{1}{2}.$$

We can thus set $\gamma = 1/2$, and by substituting this choice into (21), we obtain (25). We obtain (26) by making the same substitution into (22). ■

References

A. Agarwal and J. C. Duchi. Distributed delayed stochastic optimization. In *Advances in Neural Information Processing Systems 24*, pages 873–881. 2011. URL `http://papers.nips.cc/paper/4247-distributed-delayed-stochastic-optimization.pdf`.

H. Avron, A. Druinsky, and A. Gupta. Revisiting asynchronous linear solvers: Provable convergence rate through randomization. *IPDPS*, 2014.

A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sciences*, 2(1):183–202, 2009.

A. Beck and L. Tetruashvili. On the convergence of block coordinate descent type methods. *SIAM Journal on Optimization*, 23(4):2037–2060, 2013.

D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Pentice Hall, 1989.

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines, 2011. URL `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`.

C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, pages 273–297, 1995.

A. Cotter, O. Shamir, N. Srebro, and K. Sridharan. Better mini-batch algorithms via accelerated gradient methods. In *Advances in Neural Information Processing Systems 24*, pages 1647–1655. 2011. URL `http://papers.nips.cc/paper/4432-better-mini-batch-algorithms-via-accelerated-gradient-methods.pdf`.

J. C. Duchi, A. Agarwal, and M. J. Wainwright. Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Transactions on Automatic Control*, 57 (3):592–606, 2012.

M. C. Ferris and O. L. Mangasarian. Parallel variable distribution. *SIAM Journal on Optimization*, 4(4):815–832, 1994.

D. Goldfarb and S. Ma. Fast multiple-splitting algorithms for convex optimization. *SIAM Journal on Optimization*, 22(2):533–556, 2012.

Z. Lu and L. Xiao. On the complexity analysis of randomized block-coordinate descent methods. Technical Report arXiv:1305.4723, Simon Fraser University, 2013.

Z. Q. Luo and P. Tseng. On the convergence of the coordinate descent method for convex differentiable minimization. *Journal of Optimization Theory and Applications*, 72:7–35, 1992.

O. L. Mangasarian. Parallel gradient distribution in unconstrained optimization. *SIAM Journal on Optimization*, 33(1):916–1925, 1995.

A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19:1574–1609, 2009.

Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer Academic Publishers, 2004.

Y. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

F. Niu, B. Recht, C. Ré, and S. J. Wright. HOGWILD!: A lock-free approach to parallelizing stochastic gradient descent. *Advances in Neural Information Processing Systems 24*, pages 693–701, 2011.

Z. Peng, M. Yan, and W. Yin. Parallel and distributed sparse optimization. Preprint, 2013.

P. Richtárik and M. Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematrical Programming*, 144:1–38, 2012a.

P. Richtárik and M. Takáč. Parallel coordinate descent methods for big data optimization. Technical Report arXiv:1212.0873, 2012b.

C. Scherrer, A. Tewari, M. Halappanavar, and D. Haglin. Feature clustering for accelerating parallel coordinate descent. *Advances in Neural Information Processing Systems 25*, pages 28–36, 2012.

S. Shalev-Shwartz and T. Zhang. Accelerated mini-batch stochastic dual coordinate ascent. *Advances in Neural Information Processing Systems 26*, pages 378–385, 2013.

O. Shamir and T. Zhang. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In *Proceedings of the 30th International Conference on Machine Learning*, 2013.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1996.

P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications*, 109:475–494, 2001.

P. Tseng and S. Yun. A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming, Series B*, 117:387–423, June 2009.

P. Tseng and S. Yun. A coordinate gradient descent method for linearly constrained smooth optimization and support vector machines training. *Computational Optimization and Applications*, 47(2):179–206, 2010.

P.-W. Wang and C.-J. Lin. Iteration complexity of feasible descent methods for convex optimization. *Journal of Machine Learning Research*, 15:1523–1548, 2014.

S. J. Wright. Accelerated block-coordinate relaxation for regularized optimization. *SIAM Journal on Optimization*, 22(1):159–186, 2012.

T. Yang. Trading computation for communication: Distributed stochastic dual coordinate ascent. *Advances in Neural Information Processing Systems 26*, pages 629–637, 2013.

Journal of Machine Learning Research 16 (2015) 323-369 Submitted 9/12; Revised 9/14; Published 3/15

Geometric Intuition and Algorithms for Eν–SVM

Álvaro Barbero ALVARO.BARBERO@UAM.ES
Department of Computer Science and Knowledge–Engineering Institute
Autonomous University of Madrid
Madrid, Spain

Akiko Takeda TAKEDA@MIST.I.U-TOKYO.AC.JP
Department of Mathematical Informatics
The University of Tokyo
Tokyo, Japan

Jorge López J.LOPEZ@UAM.ES
Department of Computer Science and Knowledge–Engineering Institute
Autonomous University of Madrid
Madrid, Spain

Editor: Sathiya Keerthi

Abstract

In this work we address the Eν–SVM model proposed by Pérez–Cruz *et al.* as an extension of the traditional ν support vector classification model (ν–SVM). Through an enhancement of the range of admissible values for the regularization parameter ν, the Eν–SVM has been shown to be able to produce a wider variety of decision functions, giving rise to a better adaptability to the data. However, while a clear and intuitive geometric interpretation can be given for the ν–SVM model as a nearest–point problem in reduced convex hulls (RCH–NPP), no previous work has been made in developing such intuition for the Eν–SVM model. In this paper we show how Eν–SVM can be reformulated as a geometrical problem that generalizes RCH–NPP, providing new insights into this model. Under this novel point of view, we propose the RapMinos algorithm, able to solve Eν–SVM more efficiently than the current methods. Furthermore, we show how RapMinos is able to address the Eν–SVM model for any choice of regularization norm $\ell_{p\geq 1}$ seamlessly, which further extends the SVM model flexibility beyond the usual Eν–SVM models.

Keywords: SVM, Eν–SVM, nearest point problem, reduced convex hulls, classification

1. Introduction

Let us address the classification problem of learning a decision function f from $\mathcal{X} \subseteq \mathbb{R}^n$ to $\{\pm 1\}$ based on m training samples (X_i, y_i), with $i \in M = \{1, ..., m\}$. We assume that the training samples are i.i.d., following the unknown probability distribution $P(X, y)$ on $\mathcal{X} \times \{\pm 1\}$.

Building on the well–known support vector machine (SVM) model developed in Cortes and Vapnik (1995), a variation of it, termed ν–SVM, was proposed in Schölkopf et al. (2000) as

$$\begin{aligned} \min_{W,b,\rho,\xi} \quad & \frac{1}{2}\|W\|_2^2 - \nu\rho + \frac{1}{m}\sum_{i\in M}\xi_i \\ \text{s.t.} \quad & \begin{cases} y_i\left(W\cdot X_i + b\right) \geq \rho - \xi_i, & i \in M, \\ \xi_i \geq 0, & i \in M, \\ \rho \geq 0. \end{cases} \end{aligned} \tag{1}$$

In this formulation the value of ν is made to lie in $[0,1]$, but actually there is a value $\nu_{\min} > 0$ such that if $\nu \in [0, \nu_{\min}]$, then we obtain the trivial solution $W = b = \rho = \xi = 0$. To tackle this, Pérez-Cruz et al. (2003) proposed generalizing (1) by allowing the margin ρ to be negative and enforcing the norm of W to be unitary:

$$\begin{aligned} \min_{W,b,\rho,\xi} \quad & -\nu\rho + \frac{1}{m}\sum_{i\in M}\xi_i \\ \text{s.t.} \quad & \begin{cases} y_i\left(W\cdot X_i + b\right) \geq \rho - \xi_i, & i \in M, \\ \xi_i \geq 0, & i \in M, \\ \|W\|_2^2 = 1. \end{cases} \end{aligned} \tag{2}$$

With this modification, a non–trivial solution can be obtained even for $\nu \in [0, \nu_{\min}]$. This modified formulation was called extended–ν-SVM (Eν-SVM), and has been shown to be able to generate a richer family of decision functions, thus producing better classification results in some settings. In addition to this, Takeda and Sugiyama (2008) arrived independently to the same model by minimizing the conditional value–at–risk (CVaR) risk measure, which is often used in finance. Letting the cost function be $f(W, b, X_i, y_i) = -y_i(W \cdot X_i + b)/\|W\|$, the CVaR risk measure is defined as the mean of the $(1-\nu)$–tail distribution of f for $i \in M$ (Rockafellar and Uryasev, 2002).

One of the advantages of the ν–SVM formulation (1) comes from its multiple connections to other well–known mathematical optimization problems, some of them allowing for intuitive geometric interpretations. A schematic of such connections is presented in Figure 1. Connections 1 and 2 were introduced in the pioneer work of Bennett and Bredensteiner (2000), showing how the SVM could be interpreted geometrically. Alternatively, and following the equivalence of the SVM and ν–SVM models (connection 3, shown in Schölkopf et al., 2000), Crisp and Burges (2000) arrived to the same geometrical problem (connections 4 and 5). Such problem, known in the literature as reduced convex hull nearest–point problem (RCH–NPP), consists of finding the closest points in the reduced convex hulls of the points belonging to the positive and negative classes. This can be formulated as

$$\begin{aligned} \min_{\lambda_+,\lambda_-} \quad & \frac{1}{2}\left\| \sum_{i\in M_+}\lambda_i X_i - \sum_{i\in M_-}\lambda_i X_i \right\|_2^2 \\ \text{s.t.} \quad & \begin{cases} \sum_{i\in M_+}\lambda_i = \sum_{i\in M_-}\lambda_i = 1, \\ 0 \leq \lambda_i \leq \eta,\ i \in M, \end{cases} \end{aligned} \tag{3}$$

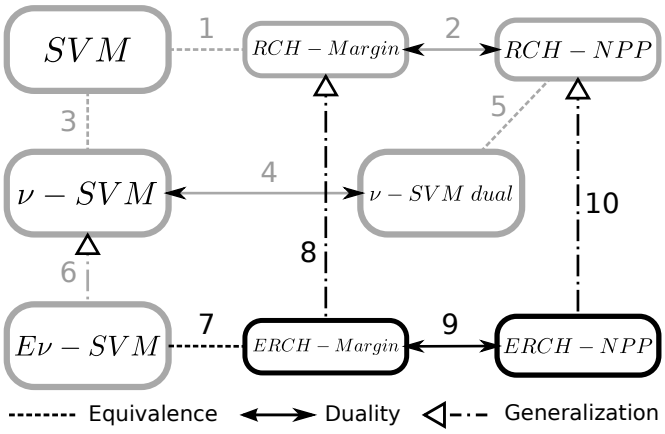


Figure 1: Relationships between the SVM, ν–SVM and other mathematical optimization problems. Connections and problems in gray were previously known, while connections and models in black are introduced in this paper.

where we denote $M_{\pm} = \{i : y_i = \pm 1\}$, and η is the reduction coefficient of the reduced convex hulls. A specific value of ν in (1) corresponds to a specific value of η in (3). Broadly speaking, the bigger ν is, the smaller η is, and the more the hulls shrink towards their barycenters.

Using the same notation, the intermediate RCH–Margin formulation in Figure 1 has the following form:

$$\min_{W,\alpha,\beta,\xi} \quad \frac{1}{2}\|W\|_2^2 + \beta - \alpha + \eta \sum_{i \in M} \xi_i \tag{4}$$

$$\text{s.t.} \quad \begin{cases} W \cdot X_i \geq \alpha - \xi_i, & i \in M_+, \\ W \cdot X_i \leq \beta + \xi_i, & i \in M_-, \\ \xi_i \geq 0, & i \in M. \end{cases}$$

At the light of these relationships and the fact that Eν–SVM is essentially a generalization of ν–SVM (connection 6, Pérez-Cruz et al., 2003), it seems natural to assume that similar connections and geometric interpretations should exist for Eν–SVM. Nevertheless, no work has been previously done along this line. Therefore, in this paper we exploit these known ν–SVM connections to develop a novel geometric interpretation for the Eν–SVM model. We will show how similar connections can be proved for Eν–SVM, and how this provides a better insight into the mathematical problem posed by this generalized model, allowing us to develop a new algorithm for Eν–SVM training.

On top of this, we demonstrate how the Eν–SVM formulation allows to extend the SVM models through the use of general $\ell_{p\geq 1}$–norm regularizations, instead of the usual ℓ_2–norm regularization. Previously, SVM models with other particular values of p have been proposed, such as ℓ_1–SVM by Zhu et al. (2003) or ℓ_∞–SVM in Bennett and Bredensteiner

(2000), acknowledging the usefulness of different ℓ_p–norms to enforce different degrees of sparsity in the model coefficients. Some work has also been done in approximating the NP–hard non–convex non–continuous ℓ_0–norm within SVM models, by methods such as iterative reweighing of ℓ_1–SVM models (Shi et al., 2011) or through expectation maximization in a Bayesian approach (Huang et al., 2009), and also in the context of least–squares support vector machines (López et al., 2011b). In spite of this, to date no efficient implementation seems to have been offered for the general $\ell_{p\geq 1}$–SVM. Similarly, no methods have been proposed either to solve an equivalent $\ell_{p\geq 1}$ version of the ERCH–NPP.

The contributions of this work on these matters are the following:

- We show how the Eν–SVM problem (2) is equivalent to an extended version of the reduced convex hull margin (RCH–Margin) problem (connections 7 and 8 in Figure 1).
- We introduce the extended reduced convex hulls nearest–point problem (ERCH–NPP), which is both a dual form of the Eν–SVM (connection 9) and a generalization of RCH–NPP (connection 10).
- For the case when the reduced convex hulls do not intersect, we show how ERCH–NPP can be reduced to the RCH–NPP problem.
- For the intersecting case we analyse how the problem becomes non–convex, and propose the RapMinos algorithm, which uses the acquired geometric insight to find a local minimum of ERCH–NPP faster than the currently available Eν–SVM solvers.
- All derivations are performed for the general $\ell_{p\geq 1}$ regularization, thus boosting the Eν–SVM model capability even further, and also providing means to solve RCH–NPP for such range of norms.
- A publicly available implementation of RapMinos is provided.

The rest of the paper is organized as follows: Section 2 describes the recasting of (2) as a geometrical problem. Section 3 shows that this geometrical problem is in fact a generalization of the standard RCH–NPP problem (3), able to find non–trivial solutions even in the case where the convex hulls intersect. In Section 4 we analyse the structure of the optimization problem posed by the ERCH–NPP problem. Based on this, Section 5 develops the RapMinos algorithm and shows its theoretical properties, while in Section 6 we present experimental results on its practical performance. Finally, Section 7 discusses briefly the results obtained and related future work.

2. Geometry in Eν-SVM

In this section we will introduce the geometric ideas behind Eν–SVM (2) by proving connections 7 and 9 in Figure 1, thus arriving to the ERCH–NPP problem. We also generalize its formulation not only to cover the ℓ_2–norm W regularization, but an arbitrary ℓ_p–norm with $p \geq 1$.

To begin with, let us define the ERCH–Margin (extended reduced–convex–hull margin) problem and its connections with Eν–SVM.

Proposition 1 *The ERCH–Margin (extended reduced–convex–hull margin) problem, defined as*

$$\min_{W:\|W\|_p=1} \quad \min_{\alpha,\beta,\xi} \quad \beta - \alpha + \eta \sum_{i\in M} \xi_i \tag{5}$$
$$s.t. \quad \begin{cases} W \cdot X_i \geq \alpha - \xi_i, & i \in M_+, \\ W \cdot X_i \leq \beta + \xi_i, & i \in M_-, \\ \xi_i \geq 0, & i \in M. \end{cases}$$

is equivalent to the Eν–SVM problem (connection 7 in Figure 1).

Proof Take (2) and multiply its objective function by $2/\nu$ [1]. Let us also consider the ℓ_p–norm, and separate the constraint $\|W\|_p = 1$ from the problem, obtaining:

$$\min_{W:\|W\|_p=1} \quad \min_{b,\rho,\xi} \quad -2\rho + \frac{2}{\nu m} \sum_{i\in M} \xi_i \tag{6}$$
$$\text{s.t.} \quad \begin{cases} y_i \left(W \cdot X_i + b\right) \geq \rho - \xi_i, & i \in M, \\ \xi_i \geq 0, & i \in M. \end{cases}$$

Denoting now $\eta = 2/(\nu m)$, $\alpha = \rho - b$ and $\beta = -\rho - b$, direct substitution makes the above problem become the ERCH–Margin problem. ■

The geometry behind this formulation is summarized in Figure 2. There we have a feasible estimate (W, α, β, ξ) which gives two parallel hyperplanes: $W \cdot X = \alpha$ and $W \cdot X = \beta$. We are seeking to optimize two conflicting goals: on the one hand we want to maximize the signed distance between both hyperplanes, given by $\alpha - \beta$, and on the other hand we want the hyperplane $W \cdot X = \alpha$ to leave as many positive points as possible to its left. The same is applicable to the hyperplane $W \cdot X = \beta$, which should leave as many negative points as possible to its right. In the configuration illustrated, preference has been given to correct classification, so that the hyperplanes "cross", and $\beta > \alpha$. Thus, the signed distance between the hyperplanes is negative in this case.

In the general case, the trade–off between these two conflicting goals is regulated by the penalty factor $\eta = 2/(\nu m)$. The slack variables ξ_i allow for errors when the hyperplanes do not leave the points to their proper side. The penalty factor keeps the errors at bay, so finally we reach a compromise between separation of the hyperplanes and correct classification.

We now move one step further and define the ERCH–NPP problem and its connection with ERCH–Margin.

Proposition 2 *The ERCH–NPP (extended reduced–convex–hull nearest–point problem) problem, defined as*

1. Note that this precludes the use of $\nu = 0$, but in practice such a value is not interesting, since (2) would only minimize the errors, which tends to overfitting.

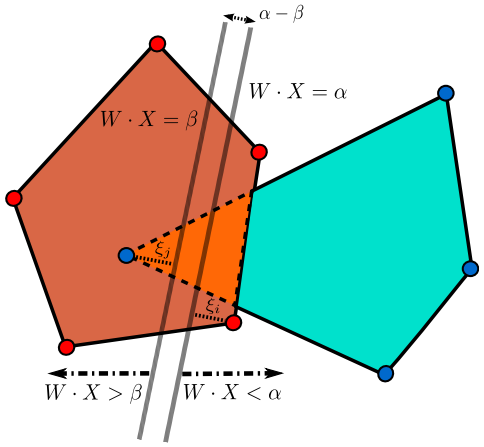


Figure 2: Illustration of the ERCH–Margin problem. The extreme positive (negative) points are printed in red (blue). The current estimate gives two parallel hyperplanes $W \cdot X = \alpha$ and $W \cdot X = \beta$ that try to separate the two classes as well as possible, while keeping far from each other. Errors are quantified by slack variables ξ_i, with two examples highlighted.

$$\min_{W:\|W\|_p=1} \quad \max_{X_+ \in \mathcal{U}_+, X_- \in \mathcal{U}_-} \quad W \cdot X_- - W \cdot X_+. \tag{7}$$

with reduced convex hulls

$$\mathcal{U}_\pm = \left\{ \sum_{i \in M_\pm} \lambda_i X_i : \sum_{i \in M_\pm} \lambda_i = 1,\ 0 \le \lambda_i \le \eta \right\},$$

is the dual problem of ERCH–Margin (connection 9 in Figure 1).

Proof The Lagrangian for the inner minimization problem in ERCH–Margin (5) reads

$$\begin{aligned} \mathcal{L} \ = \ & \beta - \alpha + \eta \sum_{i \in M} \xi_i - \sum_{i \in M_+} \lambda_i \left(W \cdot X_i - \alpha + \xi_i \right) \\ & + \sum_{i \in M_-} \lambda_i \left(W \cdot X_i - \beta - \xi_i \right) - \sum_{i \in M} \mu_i \xi_i, \end{aligned} \tag{8}$$

where we introduced the Lagrange multipliers $\lambda_i \ge 0, \mu_i \ge 0,\ i \in M$, associated to the inequality constraints of (5). Differentiating with respect to the variables being minimized and equating to zero gives

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \alpha} &= -1 + \sum_{i \in M_+} \lambda_i = 0 \quad \Rightarrow \quad \sum_{i \in M_+} \lambda_i = 1, \\
\frac{\partial \mathcal{L}}{\partial \beta} &= 1 - \sum_{i \in M_-} \lambda_i = 0 \quad \Rightarrow \quad \sum_{i \in M_-} \lambda_i = 1, \\
\frac{\partial \mathcal{L}}{\partial \xi_i} &= \eta - \lambda_i - \mu_i = 0 \quad \Rightarrow \quad 0 \leq \lambda_i \leq \eta,\ i \in M.
\end{aligned}$$

Substituting all the above in the Lagrangian (8) yields the partial dual formulation of (5):

$$\min_{W:\|W\|_p=1} \quad \max_{\lambda} \quad \sum_{i \in M_-} \lambda_i W \cdot X_i - \sum_{i \in M_+} \lambda_i W \cdot X_i \tag{9}$$
$$\text{s.t.} \quad \begin{cases} \sum_{i \in M_+} \lambda_i = \sum_{i \in M_-} \lambda_i = 1, \\ 0 \leq \lambda_i \leq \eta,\ i \in M. \end{cases}$$

Now, considering the constraints of (9) and problem (3), we are confined to the reduced convex hulls whose reduction coefficient is in this case $\eta = 2/(\nu m)$. If we have $2/(\nu m) \geq 1$, we just work in the standard convex–hulls of both subsamples. By making use of the reduced convex hulls $\mathcal{U}_\pm$ and defining $X_\pm = \sum_{i \in M_\pm} \lambda_i X_i$, problem (9) can be written more succinctly as

$$\min_{W:\|W\|_p=1} \quad \max_{X_+ \in \mathcal{U}_+, X_- \in \mathcal{U}_-} \quad W \cdot X_- - W \cdot X_+,$$

which is ERCH–NPP.

■

Once we know we are working with reduced convex hulls, further geometrical intuition can be given on what we are doing. Recall that the quantity $(W \cdot X_0 + b)/\|W\|_p$ gives the signed distance from a specific point X_0 to the hyperplane $W \cdot X + b = 0$, in terms of the ℓ_p–norm. Note that in this case we always have unitary W vectors. Since we only care about the orientation of the solution hyperplane (W, b) and not about its magnitude, problem (7) can be rewritten as

$$\max_{W,b} \quad \min_{X_+ \in \mathcal{U}_+, X_- \in \mathcal{U}_-} \quad \frac{W \cdot X_+ + b}{\|W\|_p} - \frac{W \cdot X_- + b}{\|W\|_p}, \tag{10}$$

so that we can regard that Eν-SVM finds a solution that maximizes the margin, where by "margin" we mean the smallest signed distance between the two reduced convex hulls.

There are two cases depending on the value of the reduction coefficient $2/(\nu m)$:

- If the coefficient is small enough, the reduced convex hulls will not intersect, so there exists some hyperplane W producing a perfect separation between them. Therefore, $(W^* \cdot X_+^* + b^*)/\|W^*\|_p > 0$ and $(W^* \cdot X_-^* + b^*)/\|W^*\|_p < 0$ must hold at optimality.
- If it is large enough, they will intersect, so there is no W producing perfect separation. Therefore, it is $(W^* \cdot X_+^* + b^*)/\|W^*\|_p < 0$ and $(W^* \cdot X_-^* + b^*)/\|W^*\|_p > 0$ that hold at optimality.

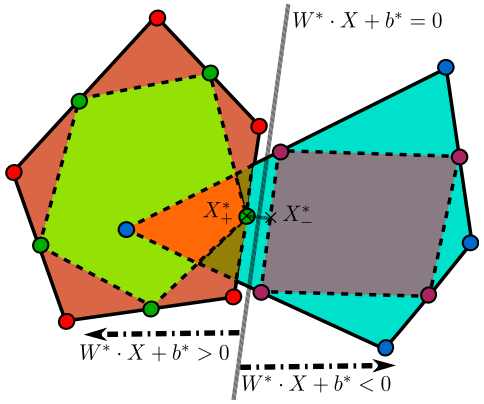


Figure 3: Case where the reduced convex hulls do not intersect ($\mu = 1/2$). The color convention is the same as in Figure 2, whereas the extreme points of the positive (negative) reduced hulls are printed in green (purple). The optimal solution is given by W^*, b^*, X_+^* and X_-^*. Observe that X_+^* (X_-^*) lies in the positive (negative) side of the hyperplane.

In the following section we will see how in the first case the problem can be reduced to the standard RCH–NPP problem, while the second case cannot be captured by such problem. This will lead to the conclusion that ERCH–NPP is a generalization of RCH–NPP (connection 10 in Figure 1), and that ERCH–Margin is a generalization of RCH–Margin (connection 8).

3. Relationship with RCH–NPP

Here we will see that ERCH–NPP (9) is in fact a generalization of RCH–NPP (3). Using the notation of the previous section, (3) can be expressed as

$$\min_{X_+ \in \mathcal{U}_+, X_- \in \mathcal{U}_-} \frac{1}{2} \|X_+ - X_-\|_q^q \quad \equiv \quad \min_{X_+ \in \mathcal{U}_+, X_- \in \mathcal{U}_-} \|X_+ - X_-\|_q \,, \tag{11}$$

where the reduction coefficient in $\mathcal{U}_\pm$ is $\eta = 2/(\nu m)$, and we again allow the use of a general ℓ_q–norm with $q \geq 1$ to measure the distance between the hulls [2].

2. While we acknowledge the interest in $q < 1$ norms in the field of Machine Learning, the use of such norms introduces an additional level of non–convexity into the problem, and thus is out of the scope of this paper.

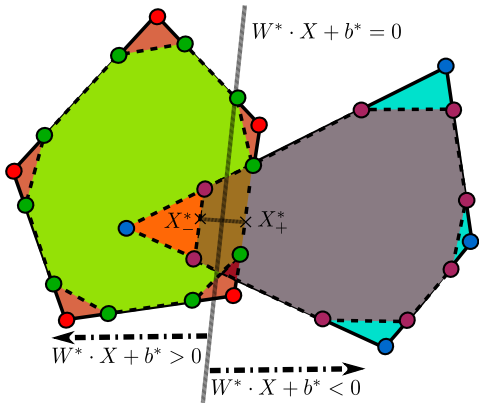


Figure 4: Case where the reduced convex hulls intersect ($\mu = 3/4$), with the same color and solution convention than in Figure 3. Observe that X_+^* (X_-^*) lies now in the negative (positive) side of the hyperplane.

If the reduction parameter $\eta = 2/(\nu m)$ is not small enough, the classes might overlap as in Figure 4, and (11) thus generates the trivial solution $X_+^* = X_-^*$, so that $W^* = 0$. The same happens with ν–SVMs, where ν must be large enough to obtain meaningful solutions. What we intend to show next is that, exactly as Eν–SVM extended ν–SVM to allow for all the range of possible values of ν (that is, $\nu \in (0, \nu_{\max}]$, with $\nu_{\max} = 2\min\{|M_+|, |M_-|\}/m$), ERCH also extends RCH to allow for all the possible values for η.

To this aim, first we show the following lemma, whose is based on the fact that if the hulls do not intersect, any solution with $\|W\|_p < 1$ is actually worse than the one obtained by trivially rescaling W so that $\|W\|_p = 1$. That is to say, relaxing the constraint in such a way does not modify the solution of the optimization, since the optimum is guaranteed to remain at the same place.

Lemma 3 *If the reduced convex hulls do not intersect, we can replace the constraint* $\|W\|_p = 1$ *in* (5) *with* $\|W\|_p \leq 1$.

Proof As was discussed above, if the reduced convex hulls do not intersect, a hyperplane W^* and a bias b^* exist such that $W^* \cdot X_+ + b^* > 0 \; \forall \; X_+ \in \mathcal{U}_+$, $W^* \cdot X_- + b^* < 0 \; \forall \; X_- \in \mathcal{U}_-$. Therefore, at the optimum of (7) and (9) the value of the inner maximum must be negative.

Since the inner problem of (9) is the dual of the inner problem of (5) and both problems are convex (linear, in fact), by strong duality the value of their objective functions is equal at the optimum (Rockafellar, 1970; Luenberger and Ye, 2008). Hence, the inner minimum of (5) must be negative as well. Therefore, for any optimal solution $(W^*, \alpha^*, \beta^*, \xi^*)$ we get the optimal objective value

$$\mathcal{P}^* = \beta^* - \alpha^* + \eta \sum_{i \in M} \xi_i^* < 0.$$

To see that we can replace the constraint $\|W\|_p = 1$ with $\|W\|_p \leq 1$, let us suppose an optimal solution $(W^*, \alpha^*, \beta^*, \xi^*)$ such that $\|W^*\|_p < 1$. We can then build another solution $(W', \alpha', \beta', \xi')$, with $W' = W^*/\|W^*\|_p$, $\alpha' = \alpha^*/\|W^*\|_p$, $\beta' = \beta^*/\|W^*\|_p$ and $\xi' = \xi^*/\|W^*\|_p$.

This solution is obviously feasible, because the constraints of (5) hold. Moreover, $\|W'\|_p = 1$ and the objective value is now

$$\mathcal{P}' = \beta' - \alpha' + \eta \sum_{i \in M} \xi'_i = \frac{\mathcal{P}^*}{\|W^*\|_p} < \mathcal{P}^*,$$

where the last inequality holds because $\mathcal{P}^* < 0$ and $\|W^*\|_p < 1$. We are minimizing in (9), so this new solution $(W', \alpha', \beta', \xi')$ is actually better than $(W^*, \alpha^*, \beta^*, \xi^*)$, which contradicts the supposed optimality of the latter. Therefore, we can safely replace $\|W\|_p = 1$ with $\|W\|_p \leq 1$. ■

The following definition and remark will also be used:

Definition 4 *The convex conjugate $\hat{f} : \hat{X} \to \mathbb{R} \cup +\infty$ of a functional $f : X \to \mathbb{R} \cup +\infty$ is $\hat{f}(\hat{x}) = \sup_{x \in X} \{\hat{x} \cdot x - f(x)\} = -\inf_{x \in X} \{f(x) - \hat{x} \cdot x\}$, where $\hat{X}$ denotes the dual space to X and the dot product operation (dual pairing) is a function $\hat{X} \times X \to \mathbb{R}$ (Rockafellar, 1970).*

Remark 5 *If $f(x) = cg(x)$, with $c > 0$ a scalar, then $\hat{f}(\hat{x}) = c\hat{g}(\hat{x}/c)$.*

Theorem 6 *The ERCH equivalent formulations* (5)–(10) *give a solution for the RCH formulation* (11) *when the reduced convex hulls do not intersect, provided that $1/p + 1/q = 1$.*

Proof By Lemma 3, we can now write problem (5) as a single minimization problem of the form

$$\min_{W,\alpha,\beta,\xi} \quad \beta - \alpha + \eta \sum_{i \in M} \xi_i \qquad (12)$$
$$\text{s.t.} \quad \begin{cases} W \cdot X_i \geq \alpha - \xi_i, & i \in M_+, \\ W \cdot X_i \leq \beta + \xi_i, & i \in M_-, \\ \xi_i \geq 0, & i \in M, \\ \|W\|_p \leq 1, \end{cases}$$

whose Lagrangian is

$$\begin{aligned} \mathcal{L} \;=\; & \beta - \alpha + \eta \sum_{i \in M} \xi_i - \sum_{i \in M_+} \lambda_i \left(W \cdot X_i - \alpha + \xi_i\right) \\ & + \sum_{i \in M_-} \lambda_i \left(W \cdot X_i - \beta - \xi_i\right) - \sum_{i \in M} \mu_i \xi_i \\ & + \delta \left(\|W\|_p - 1\right). \end{aligned}$$

However, since the ℓ_p–norm is not necessarily differentiable, we cannot proceed now as in Section 2. To derive the dual problem, we must take into account that it consists of finding the maximum, with respect to the Lagrange multipliers, of the infimum of the Lagrangian, where this infimum is with respect to the primal variables (Boyd and Vandenberghe, 2004). In our case the primal variables are W, α, β and ξ, whereas the Lagrange multipliers are λ_i, μ_i and δ. Thus, the dual of (12) translates to

$$\max_{\lambda\geq 0,\mu\geq 0,\delta\geq 0} \quad \inf_{W,\alpha,\beta,\xi} \quad \{\mathcal{L}\}\,, \tag{13}$$

where $\mathcal{L}$ is the expression above. Splitting the infimum among the different variables, we want to find

$$\begin{gathered}\inf_W \left\{\sum_{i\in M_-} \lambda_i W\cdot X_i - \sum_{i\in M_+} \lambda_i W\cdot X_i + \delta\|W\|_p\right\} + \\ \inf_\alpha \left\{-\alpha + \alpha\sum_{i\in M_+}\lambda_i\right\} + \inf_\beta \left\{\beta - \beta\sum_{i\in M_-}\lambda_i\right\} + \\ \sum_{i\in M}\inf_{\xi_i}\{\eta\xi_i - \lambda_i\xi_i - \mu_i\xi_i\} - \delta.\end{gathered}$$

For α, β and ξ we can find the infima just by differentiating and equating to 0:

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\alpha} &= \sum_{i\in M_+}\lambda_i - 1 = 0 \quad\Rightarrow\quad \sum_{i\in M_+}\lambda_i = 1,\\ \frac{\partial\mathcal{L}}{\partial\beta} &= 1 - \sum_{i\in M_-}\lambda_i = 0 \quad\Rightarrow\quad \sum_{i\in M_-}\lambda_i = 1,\\ \frac{\partial\mathcal{L}}{\partial\xi_i} &= \eta - \lambda_i - \mu_i = 0 \quad\Rightarrow\quad 0\leq\lambda_i\leq\eta,\ \ i\in M.\end{aligned}$$

As for W, we can write the infimum as $\inf\left\{-\sum_{i\in M}\lambda_i y_i W\cdot X_i + \delta\|W\|_p\right\}$. This expression follows the form of the convex conjugate as presented in Definition 4, where we can identify the functional $f(W) = \delta\|W\|_p$ and the dual variable $\hat{W} = \sum_{i\in M}\lambda_i y_i X_i$. Moreover, assuming for the moment that $\delta > 0$ and using Remark 5, we have $g(W) = \|W\|_p$.

Since the convex conjugate of the ℓ_p–norm is given by

$$\hat{g}(\hat{W}) \;=\; \begin{cases}0 & \text{if } \|\hat{W}\|_q\leq 1,\\ +\infty & \text{otherwise,}\end{cases}$$

where $1/p + 1/q = 1$ (see Boyd and Vandenberghe, 2004), we get in our case that the term $\inf\left\{-\sum_{i\in M}\lambda_i y_i W\cdot X_i + \delta\|W\|_p\right\}$ equals

$$-\hat{f}\left(\hat{W}\right) = -\delta\hat{g}\left(\frac{\hat{W}}{\delta}\right) \;=\; \begin{cases}0 & \text{if } \left\|\frac{\hat{W}}{\delta}\right\|_q\leq 1,\\ -\infty & \text{otherwise,}\end{cases}$$

which can be rewritten as

$$-\delta\hat{g}\left(\frac{1}{\delta}\sum_{i\in M}\lambda_i y_i X_i\right) \;=\; \begin{cases} 0 & \text{if } \left\|\sum_{i\in M}\lambda_i y_i X_i\right\|_q \leq \delta, \\ -\infty & \text{otherwise.} \end{cases}$$

The optimum will be located in the region where the convex conjugate is finite, so (13) is equivalent to

$$\begin{aligned} \max_{\lambda,\delta} \quad & -\delta \\ \text{s.t.} \quad & \begin{cases} \sum_{i\in M_+}\lambda_i = \sum_{i\in M_-}\lambda_i = 1, \\ 0 \leq \lambda_i \leq \eta, \;\; i \in M, \\ \left\|\sum_{i\in M}\lambda_i y_i X_i\right\|_q \leq \delta. \end{cases} \end{aligned} \tag{14}$$

On the other hand, when $\delta = 0$ the infimum on W is just $\inf\left\{-\sum_{i\in M}\lambda_i y_i W\cdot X_i\right\}$. Differentiating with respect to W we obtain that $X_+ = X_-$, so that $W = 0$. Consequently, $\left\|\sum_{i\in M}\lambda_i y_i X_i\right\|_q = 0 = \delta$, which satisfies (14).

Observe that the non–negativity constraints of the Lagrange multipliers in (13) are subsumed in the constraints above. This can be further rewritten, removing δ, as

$$\begin{aligned} \min_{\lambda} \quad & \left\|\sum_{i\in M}\lambda_i y_i X_i\right\|_q \\ \text{s.t.} \quad & \begin{cases} \sum_{i\in M_+}\lambda_i = \sum_{i\in M_-}\lambda_i = 1 \\ 0 \leq \lambda_i \leq \eta, \;\; i \in M, \end{cases} \end{aligned}$$

that is, problem (11).

■

Therefore, when the hulls do not intersect, ERCH–NPP results in the standard RCH–NPP problem. It is worth noting that Bennett and Bredensteiner (2000) already described how RCH–NPP relates to RCH–Margin (which is a particular case of our ERCH–Margin formulation 5 for non–intersecting hulls), for the ℓ_1, ℓ_2 and ℓ_∞–norms. Nevertheless, their proof was omitted due to space constraints. We cover general p and q, which include all these as particular cases.

Addressing now the non–intersecting case, we introduce another lemma, analogous to Lemma 3.

Lemma 7 *If the reduced convex hulls intersect, we can replace the constraint* $\|W\|_p = 1$ *in* (5) *with* $\|W\|_p \geq 1$.

Proof Just follow a similar argument to the one presented for Lemma 3. By the nature of problem (7), and since there is overlap, we obtain $W^* \cdot X_+^* + b^* < 0$ and $W^* \cdot X_-^* + b^* > 0$ for any optimal W^*, X_+^*, X_-^* (see Figure 4).

Therefore, at the optimum of (7) and (9) the value of the inner maximum must be positive. Supposing that $\|W^*\|_p > 1$ allows us to build an alternative feasible solution $(W^*/\|W^*\|_p, \alpha^*/\|W^*\|_p, \beta^*/\|W^*\|_p, \xi^*/\|W^*\|_p)$, whose norm is unitary and whose primal value is less than that of our hypothetical optimal solution, contradicting thus this optimality. ■

The problem can be then rewritten as

$$\min_{W} \min_{\alpha,\beta,\xi} \quad \beta - \alpha + \eta \sum_{i \in M} \xi_i$$
$$\text{s.t.} \quad \begin{cases} W \cdot X_i \geq \alpha - \xi_i, & i \in M_+, \\ W \cdot X_i \leq \beta + \xi_i, & i \in M_-, \\ \xi_i \geq 0, & i \in M, \\ \|W\|_p \geq 1. \end{cases}$$

In contrast to the derivation in Theorem 6, obtaining the dual of this problem is counterproductive. Since the constraint $\|W\|_p \geq 1$ is non–convex, a non–zero dual gap is bound to appear. Therefore, solving the dual problem would only provide an approximate solution to the ERCH. Instead of following such a derivation, we take the ERCH–NPP formulation in 7 and plug in the modified constraint on W, obtaining

$$\min_{\|W\|_p \geq 1} \quad \max_{X_+ \in \mathcal{U}_+, X_- \in \mathcal{U}_-} \quad W \cdot X_- - W \cdot X_+.$$

The immediate advantage of this formulation of the ERCH is that, whatever the data points X, a trivial solution $W = 0$ is never obtained. In comparison, the RCH–NPP model always produces the trivial solution whenever the reduced hulls intersect. Joining this and the facts above, it is immediate that ERCH–NPP can be regarded as a generalization of RCH–NPP.

Theorem 8 *ERCH–NPP is a generalization of RCH–NPP (connection 10 in Figure 1).*

Proof Given the data points for which to solve ERCH–NPP, the reduced convex hulls formed by such points might or might not intersect. If they do not intersect, by Theorem 6 the solution of the ERCH–NPP problem is exactly the solution of RCH–NPP. If they do intersect, then RCH–NPP fails to find a non–trivial solution, while ERCH–NPP does not, by Lemma 7. Therefore, ERCH–NPP covers all feasible cases for RCH–NPP plus a new set, hence being a generalization of RCH–NPP. ■

Note that ERCH–Margin in (5) is nothing but RCH–Margin in (4), with the additional requirement $\|W\|_p = 1$. Regarding the above two possible cases, we have seen that if the

reduced convex hulls do not intersect we can substitute this constraint with $\|W\|_p \leq 1$, so that we obtain the solution of RCH–NPP and, by strong duality, that of RCH–Margin. When they do intersect, RCH–NPP and RCH–Margin give a trivial 0 solution, whereas ERCH–NPP and ERCH–Margin do not, since we can use the constraint now that $\|W\|_p \geq 1$. Thus, it can be stated as follows:

Corollary 9 *ERCH–Margin is a generalization of RCH–Margin (connection 8 in Figure 1).*

4. Structure of the ERCH–NPP

The actual problem of solving ERCH–NPP

$$\min_{W:\|W\|_p=1} \quad \max_{X_+\in\mathcal{U}_+, X_-\in\mathcal{U}_-} \quad W\cdot X_- - W\cdot X_+,$$

is non–trivial, the main reason being that the constraint $\|W\|_p = 1$ imposes a non–convex feasible set. This might lead to local minima among other issues, which in turn make the optimization process difficult.

As described in the previous section, if the reduced convex hulls for the given data points do not intersect, then the problem above can be reduced to the standard RCH–NPP. Therefore, in such case the optimization can be performed by just employing one of the available solvers for RCH–NPP, such as the RCH–SK and RCH–MDM methods proposed respectively in Mavroforakis and Theodoridis (2006) and López et al. (2011a).

Of course, such methods cannot be applied in the intersecting hulls case, which is actually the one of most interest, since it cannot be addressed by the RCH–NPP model. It is therefore necessary to develop an optimization algorithm suitable for the general ERCH–NPP case; to do so we will first analyze the structure of the optimization problem posed by ERCH–NPP.

It is clear that we can recast the problem to solve as the minimization of a function

$$\min_{\|W\|_p=1} \quad f(W), \tag{15}$$

where

$$\begin{aligned} f(W) &= \max_{X_+\in\mathcal{U}_+, X_-\in\mathcal{U}_-} \{W\cdot X_- - W\cdot X_+\}, \\ &= \max_{X_-\in\mathcal{U}_-} \{W\cdot X_-\} - \min_{X_+\in\mathcal{U}_+} \{W\cdot X_+\}. \end{aligned} \tag{16}$$

This can be further rewritten in the following form

$$f(W) = \max_{X\in\mathcal{M}} W\cdot X, \tag{17}$$

where $\mathcal{M}$ is the Minkowski polygon of the data, which is obtained through the Minkowski difference $\mathcal{M} = \mathcal{U}_- \ominus \mathcal{U}_+$ defined as the set

$$X \ominus Y \equiv \{z | z = x - y, x \in X, y \in Y\}.$$

The Minkowski polygon has been used historically in the context of RCH–NPP to design efficient solvers (Mavroforakis et al., 2007; Keerthi et al., 2000). The properties of the Minkowski difference guarantee that the difference of two convex sets is also a convex set (Ericson, 2005), and so in our problem $\mathcal{M}$ fancies this property. In this paper we will exploit both representations (16) and (17) to take advantage of the structure of the problem.

Interestingly, the maximum and minimum in Equation (16) can be obtained efficiently from the observations in the work of Mavroforakis and Theodoridis (2006) about the extreme points of reduced convex hulls. As they show, any extreme point in a reduced convex hull can be expressed in the form

$$X_E = \sum_{i=1}^{\lfloor 1/\eta \rfloor} \eta X_i + (1 - \lfloor 1/\eta \rfloor \eta)\, X_{\lceil 1/\eta \rceil},$$

that is, the convex combination of $\lceil 1/\eta \rceil$ points, where $\lfloor 1/\eta \rfloor$ of them are given a weight of η and an additional one the remaining weight $1 - \lfloor 1/\eta \rfloor \eta$ (if it is non–zero). Using this property, they note that the extreme points with minimum margin for a given W can be found as

$$\arg\min_{X \in \mathcal{U}} \{W \cdot X\} \;=\; \sum_{i=1}^{\lfloor 1/\eta \rfloor} \eta X_i^{inc} + (1 - \lfloor 1/\eta \rfloor \eta)\, X_{\lceil 1/\eta \rceil}^{inc},$$

where the X_i^{inc} are the original points X_i sorted increasingly by their margin values

$$W \cdot X_1^{inc} \;\leq\; W \cdot X_2^{inc} \;\leq \ldots \leq\; W \cdot X_N^{inc}.$$

These observations can also be applied here to find the value of $f(W)$, as

$$\arg\max_{X_- \in \mathcal{U}_-} \{W \cdot X_-\} \;=\; \sum_{i=1}^{\lfloor 1/\eta \rfloor} \eta X_{i_-}^{dec} + (1 - \lfloor 1/\eta \rfloor \eta)\, X_{\lceil 1/\eta \rceil_-}^{dec}, \tag{18}$$

$$\arg\min_{X_+ \in \mathcal{U}_+} \{W \cdot X_+\} \;=\; \sum_{i=1}^{\lfloor 1/\eta \rfloor} \eta X_{i_+}^{inc} + (1 - \lfloor 1/\eta \rfloor \eta)\, X_{\lceil 1/\eta \rceil_+}^{inc}, \tag{19}$$

where the X_-^{dec} are the points from the negative class sorted by margin decreasingly, and the X_+^{inc} are the points from the positive class sorted by margin increasingly:

$$W \cdot X_{1_-}^{dec} \;\geq\; W \cdot X_{2_-}^{dec} \;\geq \ldots \geq\; W \cdot X_{m_-}^{dec},$$
$$W \cdot X_{1_+}^{inc} \;\leq\; W \cdot X_{2_+}^{inc} \;\leq \ldots \leq\; W \cdot X_{m_+}^{inc}.$$

The computation of $f(W)$, hence, can be easily done by just performing these sortings, which only require $O(m \log(m))$ operations. This ability to find the value of $f(W)$ for a fixed W is the key for computing the gradient of $f(W)$. Supposing $Z_+ = \arg\min_{X_+ \in \mathcal{U}_+} \{W \cdot X_+\}$ and $Z_- = \arg\max_{X_- \in \mathcal{U}_-} \{W \cdot X_-\}$ and that both Z_+ and Z_- are singletons (no other choices of $X_\pm$ attain the minimum/maximum values), the gradient is clearly $\nabla f(W) = \frac{\partial}{\partial W}(W \cdot Z_- - W \cdot Z_+) = Z_- - Z_+$.

It might happen, however, that Z_+ or Z_- (or both) is a set of points instead of a singleton. If that is the case, which takes place in practice quite often, a set of gradients are possible, constituting the subdifferential

$$\begin{aligned}\frac{\partial f}{\partial W} &= \frac{\partial}{\partial W}\left(\max_{X_-\in\mathcal{U}_-}\{W\cdot X_-\} - \min_{X_+\in\mathcal{U}_+}\{W\cdot X_+\}\right),\\ &= \frac{\partial}{\partial W}\max_{X_-\in\mathcal{U}_-}\{W\cdot X_-\} - \frac{\partial}{\partial W}\min_{X_+\in\mathcal{U}_+}\{W\cdot X_+\},\\ &= \frac{\partial}{\partial W}\max_{X_-\in\mathcal{U}_-}\{W\cdot X_-\} + \frac{\partial}{\partial W}\max_{X_+\in\mathcal{U}_+}\{-W\cdot X_+\}.\end{aligned}$$

Invoking the property that the subdifferential of the maximum of a set of convex functions (linear, in this case) at a given point is the convex hull of the subdifferentials of the functions attaining such maximum at that point (Boyd and Vandenberghe, 2007) [3], we obtain that

$$\begin{aligned}\frac{\partial f}{\partial W} &= \operatorname{conv}\left\{X \,\middle|\, X\cdot W = \max_{X_-\in\mathcal{U}_-} W\cdot X_-\right\}\\ &\quad - \operatorname{conv}\left\{X \,\middle|\, X\cdot W = \min_{X_+\in\mathcal{U}_+} W\cdot X_+\right\},\end{aligned} \tag{20}$$

where conv stands for standard convex hull.

A more intuitive way to understand this subdifferential is to note that the orderings X_-^{dec} and X_+^{inc} need not be unique, since it might well happen that, for instance, $W\cdot X_{i_-}^{dec} = W\cdot X_{(i+1)_-}^{dec}$, and so the relative position of these two elements in the ordering is arbitrary. For these multiple orderings the assignment of weights to obtain $Z_- = \arg\max_{X_-\in\mathcal{U}_-}\{W\cdot X_-\}$ can produce a set of different Z_- vectors, thus explaining the non–singleton subdifferential. Note however that not every reordering produces a different subgradient, since as shown in equations (18-19) the $\lfloor 1/\eta\rfloor$ first $X_{i_\pm}$ vectors in the orderings receive all the same weight η, while all the vectors from the $\lceil 1/\eta\rceil + 1$ have no weight in the combination. In particular, swaps in the ordering of two vectors $W\cdot X_{i_\pm} = W\cdot X_{(i+1)_\pm}$ with equal weight in such combination produce no change in the resulting subgradient. Therefore, only equalities involving the $X_{\lceil 1/\eta\rceil_\pm}$ vector can produce different subgradients. These observations will become useful when discussing the stepsize selection of our proposed algorithm (Section 5.4).

With the subdifferential at hand, one could easily design a subgradient projection (SP) method (Bertsekas, 1995) to solve problem (15). For clarity of the explanations to follow, an outline of this method for the minimization of a general function $f(x)$ constrained to some set X is presented as Algorithm 1. As detailed in the pseudocode, the algorithm basically alternates update steps and projection steps. In the former, the current estimate of the solution is updated by following the negative of some subgradient belonging to the subdifferential, while in the latter the updated solution is moved back to the feasible region

3. This property can be inferred from the observations in Clarke (1990, p. 10–11).

Algorithm 1 Subgradient Projection (SP) method for $\min_{x \in X} f(x)$

Initialization: chose $x^0 \in X$, $t = 0$.
while stopping criterion not met **do**
  Compute a subgradient g^t of $f(x^t)$.
  Select an updating stepsize s^t.
  Update step: $z^{t+1} = x^t - s^t g^t$.
  Projection step: $x^{t+1} = P\left[z^{t+1}\right]_X$
  $t \leftarrow t + 1$.
end while
return x^t.

through an Euclidean projection. This method, though fairly simple, is bound to perform poorly, since it uses little information about the problem at hand. Furthermore, due to the non–convex nature of the problem it is not easy to give any guarantees on convergence.

In spite of SP presenting these drawbacks, we show here how building on top of it and introducing adaptations for this particular problem, it is able to find a solution for ERCH–NPP efficiently. We enhance the SP algorithm by modifying its four basic operations: the computation of the updating direction, the updating stepsize selection, the projection operator, and the initialization procedure.

To guide such modifications, we first introduce the following theorem, which forms the base of our algorithm:

Theorem 10 *The optimum of ERCH-NPP when the reduced hulls intersect is located at a non–differentiable point.*

The details of the proof for this theorem are not relevant for the discussion to follow, so it is relegated to the Appendix. Its importance rather stems from the fact that we can guide the optimization procedure to look just for non–differentiable points in the search space, and still be able to reach the optimum.

5. The RapMinos Algorithm

We describe now the distinctive elements of our proposed solver for ERCH-NPP: the RAdially Projected MInimum NOrm Subgradient (RapMinos) algorithm.

5.1 Updating Direction

The first thing to adapt is the direction used for the update. Using the negative of an arbitrary subgradient, as in SP, can result in non–decreasing updating directions (Bertsekas, 1995), which in turn can make hard to provide any guarantees on convergence. Therefore, we introduce a modification that guarantees descent in the objective function in every iteration, and also allows to perform optimality checks easily. To do so, we need to resort to the concept of minimum–norm subgradient (MNS) from the literature of non–smooth optimization (Clarke, 1990):

Definition 11 *Consider a non–smooth function $f(x)$, and its subdifferential set $\partial f(x)$ at a point x. The minimum–norm subgradient $g^*(x)$ is then*

$$g^*(x) = \underset{g \in \partial f(x)}{\arg\min} ||g||,$$

for some proper norm $||\cdot||$.

In an unconstrained problem, the direction given by $d = -g^*(x)$ is guaranteed to be a descent direction. When constraints are introduced, however, such a guarantee is harder to obtain. We nevertheless are able to meet it through the following theorem:

Theorem 12 (Descent directions for ERCH) *Consider the Lagrangian of problem (15)*

$$L(W, \lambda) = f(W) + \lambda(||W||_p - 1),$$

with $\lambda \in \mathbb{R}$ the Lagrange coefficient. Now consider the subdifferential set of the Lagrangian,

$$\Gamma(W) = \partial f(W) + \lambda \partial ||W||_p,$$

and suppose that the current W is feasible, so that $||W||_p = 1$. Then the element with minimum norm in $\Gamma(W)$,

$$\gamma^*(W) = \underset{\gamma \in \Gamma(W)}{\arg\min} ||\gamma||, \tag{21}$$

meets $||\gamma^*(W)|| = 0$ if W is a local minimum of the problem. Else, the direction $d = -\gamma^*(W)$ is guaranteed to be a descent direction.

Once again, the proof of the theorem is relegated to the Appendix to avoid technical clutter in the discussion. The theorem itself provides a powerful tool to obtain both descent directions and a reliable check for optimality, as we will see. But of course, a procedure must be devised to find the appointed MNS of the Lagrangian in (21). A helpful observation for doing so is the fact that the optimal value of the Lagrange coefficient λ can be determined in closed form. Consider (21), and observe that the diversity in the set $\Gamma(W)$ is given by the possible elements of the subdifferentials $\partial f(W)$ and $\partial ||W||_p$, and the Lagrange coefficient λ. To simplify notation, let us define $g \in \partial f(W)$, $n \in \partial ||W||_p$ elements of the subdifferentials. By considering the problem just in terms of λ we can write

$$\begin{aligned} & \min_{\lambda} ||g + \lambda n||_2^2, \\ = \; & \min_{\lambda} ||g||_2^2 + \lambda^2 ||n||_2^2 + 2\lambda g \cdot n. \end{aligned}$$

Note that even if the MNS is defined for any proper norm, we have employed the ℓ_2–norm here to ease the calculations. Computing now the derivative and solving for λ we obtain

$$\begin{aligned} \frac{\partial}{\partial \lambda} &= 2\lambda ||n||_2^2 + 2g \cdot n = 0, \\ \lambda^* &= -\frac{g \cdot n}{n \cdot n} = -P\left[g\right]_n, \end{aligned}$$

which is precisely the negative of the coefficient for the Euclidean projection of g on n, $P\left[g\right]_n$. With this in mind and assuming that the subdifferential $\partial ||W||_p$ is a singleton n [4], problem (21) is simplified down to

$$\begin{aligned} \min_g \quad & ||g - P\left[g\right]_n \cdot n||_2^2, \\ \text{s.t.} \quad & g \in \partial f(W), \end{aligned} \tag{22}$$

and the resulting updating direction d would be $d = -(g^* - P\left[g^*\right]_n \cdot n)$ with g^* the minimizer of the problem. Before discussing how this minimizer is found, first we show how the computation of the vector $n = \partial ||W||_p$ is performed.

Since we have assumed that $||W||_p = 1$, we can safely temporarily replace the constraint by $||W||_p^p = 1$, which eases the calculations. The derivative is then

$$\frac{\partial ||W||_p^p}{\partial W} = \frac{\partial}{\partial W} \sum_i |W_i|^p .$$

It is easier to develop this derivative by considering each entry of the gradient vector separately,

$$\begin{aligned} \left[\frac{\partial ||W||_p^p}{\partial W}\right]_k &= \frac{\partial}{\partial W_k} \sum_i |W_i|^p , \\ &= p\,|W_k|^{p-1} \frac{\partial}{\partial W_k} |W_k|, \\ &= p\,|W_k|^{p-1} \operatorname{sign}(W_k), \end{aligned} \tag{23}$$

where the subgradient $\frac{\partial}{\partial W}|W_i|$ is the sign function

$$\operatorname{sign}(x) = \begin{cases} 1 & \text{if} \quad x > 0, \\ -1 & \text{if} \quad x < 0, \\ 0 & \text{if} \quad x = 0. \end{cases}$$

A few technicalities have been omitted in this derivation: we refer to the Appendix for the details.

Now that we have a way to compute n, we show how to find the minimizer g^* of problem (22). This is easy to do upon realizing that it can be rewritten as a modified standard RCH-NPP. To do so, first observe that

4. This is not met for the particular cases of norms $p = 1$ and $p = \infty$, as they present non–differentiable points. However, taking this assumption produces no harm in practice. Refer to the Appendix for further discussion on this issue.

$$\begin{aligned} g - P\,[g]_n \cdot n &= g - \frac{g \cdot n}{n \cdot n} n, \\ &= g - \frac{nn^T}{n \cdot n} g, \\ &= \left(\mathcal{I} - \frac{nn^T}{n \cdot n}\right) g, \\ &= \mathcal{N} g, \end{aligned}$$

where $\mathcal{I}$ is the identity matrix and $\mathcal{N} = \mathcal{I} - \frac{nn^T}{n \cdot n}$ transformation matrix. The problem then becomes

$$\begin{aligned} \min_{g} \quad & ||\mathcal{N} g||_2^2 \,, \\ \text{s.t.} \quad & g \in \partial f(W). \end{aligned} \tag{24}$$

To realize the underlying connections with RCH–NPP, we shall rewrite explicitly the constraint $g \in \partial f(W)$. To do so, remember that g can be expressed as the difference of two extreme points (see Eq. 20) and these in turn as a convex combination of the data in each class (Eqs. 18 and 19). Therefore we have that $g = \sum_{i \in M_-} \mu_i X_i - \sum_{i \in M_+} \mu_i X_i$ for some combination weights μ_i, which should be set according to the margin orderings (as explained in Eqs. 18-19). To be more precise, let us define the index sets

$$\begin{aligned} \mathcal{S}_+ &= \left\{ i \;\middle|\; i \in M_+ \,,\; W \cdot X_i = W \cdot X^{inc}_{\lceil 1/\eta \rceil_+} \right\}, \\ \mathcal{S}_- &= \left\{ i \;\middle|\; i \in M_- \,,\; W \cdot X_i = W \cdot X^{dec}_{\lceil 1/\eta \rceil_-} \right\}, \\ \mathcal{Q}_+ &= \left\{ i \;\middle|\; i \in M_+ \,,\; W \cdot X_i < W \cdot X^{inc}_{\lceil 1/\eta \rceil_+} \right\}, \\ \mathcal{Q}_- &= \left\{ i \;\middle|\; i \in M_- \,,\; W \cdot X_i > W \cdot X^{dec}_{\lceil 1/\eta \rceil_-} \right\}. \end{aligned}$$

These sets can be explained as follows. At a differentiable point W the orderings X^{inc}_+ and X^{dec}_- are unique, and so $\mathcal{S}_\pm$ is a singleton containing just the index corresponding to $X^{inc/dec}_{\lceil 1/\eta \rceil_\pm}$, which is the only pattern with weight $(1 - \lfloor 1/\eta \rfloor\, \eta)$, while the sets $\mathcal{Q}_\pm$ contain the indices of all patterns with weight η. At a non–differentiable point, however, the sets $\mathcal{S}_\pm$ contain the indices of those patterns that can be swapped in the ordering while keeping the same objective value in (7), since they have equal margin. While the patterns indexed by $\mathcal{Q}_\pm$ still maintain a fixed weight η, the weights of the patterns indexed by $\mathcal{S}_\pm$ can be rearranged to obtain different subgradients. We are able to represent implicitly the whole subdifferential with $\mathcal{S}_\pm$ and $\mathcal{Q}_\pm$. Indeed, we can define the constant $C = \sum_{i \in \mathcal{Q}_-} \eta X_i - \sum_{i \in \mathcal{Q}_+} \eta X_i$, which only contains fixed terms, and rewrite our direction problem (24) as

$$\begin{aligned} \arg\min_{\mu} \quad & \left\| \mathcal{N}\left(C + \sum_{i\in\mathcal{S}_-} \mu_i X_i - \sum_{i\in\mathcal{S}_+} \mu_i X_i \right) \right\|_2^2, \\ \text{s.t.} \quad & \begin{cases} \sum_{i\in S_-} \mu_i + \sum_{i\in Q_-} \eta = 1, \\ \sum_{i\in S_+} \mu_i + \sum_{i\in Q_+} \eta = 1, \\ 0 \le \mu_i \le \eta, \quad \forall\, i \in S_\pm. \end{cases} \end{aligned}$$

Note that the constraints are nothing but the RCH–NPP constraints (problem 3), though taking into account that the points in the $Q_\pm$ sets have fixed weight η. Using this fact and defining $\tilde{X} = \mathcal{N}X$, $\tilde{C} = \mathcal{N}C$ we get the simplified problem

$$\begin{aligned} \arg\min_{\mu_i, i\in S_\pm} \quad & \left\| \tilde{C} + \sum_{i\in\mathcal{S}_-} \mu_i \tilde{X}_i - \sum_{i\in\mathcal{S}_+} \mu_i \tilde{X}_i \right\|_2^2, \\ \text{s.t.} \quad & \begin{cases} \sum_{i\in S_-} \mu_i + |Q_-|\eta = 1, \\ \sum_{i\in S_+} \mu_i + |Q_+|\eta = 1, \\ 0 \le \mu_i \le \eta, \quad \forall\, i \in S_\pm, \end{cases} \end{aligned} \tag{25}$$

where only the μ weights of the non–fixed points in $\mathcal{S}_\pm$ need to be optimized over. This problem is solved trivially by introducing some small modifications into an RCH–NPP solver; more details on this are given in the implementation section (6.1). The relevant fact here is that we can obtain a descent direction in our ERCH algorithm by solving problem (25), and this can be done efficiently by invoking an RCH solver.

5.2 Geometric Intuition of Updating Direction

Even though involved arguments from non–smooth optimization have been used to obtain the updating direction, it turns out that an easy geometric intuition can be given for it. But before introducing it, some definitions from geometry are needed:

Definition 13 *Supporting hyperplane: given a set $X \in \mathbb{R}^n$, a hyperplane h_X supports X if X is entirely contained in one of the two closed half–spaces determined by $h_X(x)$ and h_X contains at least one point from X.*

Definition 14 *Supporting hyperplane at a point: given a closed set $X \in \mathbb{R}^n$ and a point x in the boundary of X, a hyperplane $h_X(x)$ supports X at x if it is supports X and contains x. If the set X is convex, $h_X(x)$ is guaranteed to exist (Boyd and Vandenberghe, 2004).*

We further introduce the definition of supporting projection as

Definition 15 *Supporting projection: given a closed convex set $X \in \mathbb{R}^n$, a point $x \in X$ at a boundary of X and a vector v originating at x, we define the supporting projection of v on*

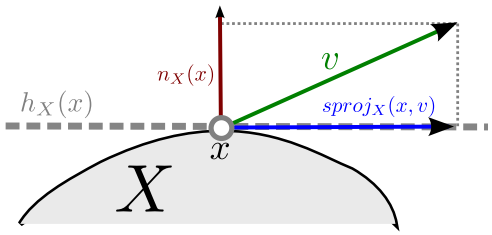


Figure 5: Depiction of the geometric concepts of supporting hyperplane at a point and of supporting projection. The hyperplane $h_X(x)$ supports the set X at the point x. The supporting projection of v is then obtained by projecting v onto $h_X(x)$, which is equivalent to removing from v its projection on the normal vector $n_X(x)$.

X, $\text{sproj}_X(x, v)$ *as the Euclidean projection of* v *on the supporting hyperplane at* x, $h_X(x)$. *That is to say*

$$\begin{aligned} \underset{X}{\text{sproj}}(x, v) &= P\left[v\right]_{h_X(x)} = v - P\left[v\right]_{n_X(x)} n_X(x), \\ &= v - \frac{v \cdot n_X(x)}{n_X(x) \cdot n_X(x)} n_X(x), \end{aligned} \tag{26}$$

for $n_X(x)$ *the normal vector defining* $h_X(x)$. *This is equivalent to removing from* v *its projection on the normal vector* $n_X(x)$.

An illustrating example on these concepts is given in Figure 5.

Using these, we can see that our updating direction takes the form

$$d = -(g^* - P\left[g^*\right]_n n) = -\underset{||W||_p=1}{\text{sproj}}\ (W, g^*), \tag{27}$$

since the normal vector $n_{||W||_p=1}(W)$ is nothing but the derivative $\partial||W||_p = n$. That is to say, our proposed direction follows the negative of the supporting projection of g^*, with g^* the subgradient that produces the smallest such projection.

5.3 Projection Operator

Now that the updating direction is well defined, we move on to defining a suitable projection operator, which is required to meet our assumption above about W being feasible at every iteration ($||W||_p = 1$). Instead of using Euclidean projection, as is the rule in SP, we instead employ radial projection on the ℓ_p unit–ball (Figueiredo and Karlovitz, 1967), which is defined as

$$R_p\left[x\right] = \begin{cases} x & \text{if} \quad ||x||_p \leq 1, \\ x/||x||_p & \text{if} \quad ||x||_p > 1. \end{cases} \tag{28}$$

One major advantage of using this operator instead of Euclidean projections is its simplicity and generality for any norm p. Furthermore we have the following property:

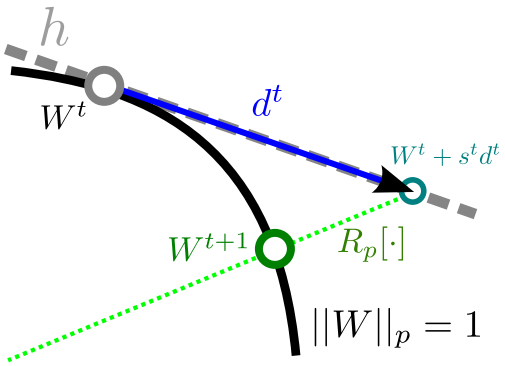


Figure 6: Example of an updating step within the RapMinos algorithm. The point W^t is updated by a displacement along the supporting hyperplane $h = h_{||W||_p=1}(W^t)$ following direction d^t, and then mapped back to the feasible region by means of a radial projection.

Lemma 16 *The radial projection $R_p[x]$ never increases the ℓ_p norm of x, i.e., $||R_p[x]||_p \leq ||x||_p$.*

Proof This is immediate from the definition, since for $||x||_p \leq 1$ the projection leaves x unchanged, and for $||x||_p \geq 1$, $R_p[x] = x/||x||_p$, and so $||R_p[x]||_p = ||x/||x||_p||_p = ||x||_p/||x||_p = 1 \leq ||x||_p$. ■

It must be noted that applying this projection operator to the ERCH–NPP problem could, in principle, lead to infeasible W values, since for $||W||_p < 1$ the projection leaves W unchanged, i.e., $R_p[W] = W$. This violates the constraint $||W||_p = 1$, producing an infeasible W at the end of the iteration. Fortunately, it is easy to show that this situation cannot happen during our algorithm.

Lemma 17 *For a given W^t vector with $||W^t||_p = 1$ and any stepsize $s^t \in \mathbb{R}$, the update $W^{t+1} = R_p\left[W^t + s^t d^t\right]$ with d^t as defined in (27) meets $||W^{t+1}||_p = 1$.*

Proof The proof follows from the fact that the displaced point $W^t + s^t d^t$ is guaranteed to lie in the supporting hyperplane $h_{||W||_p=1}(W^t)$, given the nature of the updating direction d^t and the fact that the $||W^t||_p = 1$, i.e., W^t lies in the border of the convex set $||W^t||_p \leq 1$ (see Figure 6). Because of the properties of a supporting hyperplane, every point in h is guaranteed to be outside or in the border of the set $||W^t||_p \leq 1$, and so $||W^t + s^t d^t||_p \geq 1$. Therefore, using the definition of radial projection, $||W^{t+1}||_p = ||R_p\left[W^t + s^t d^t\right]||_p = 1$. ■

Thus, we are guaranteed to remain in the feasible set throughout the whole algorithm as long as $||W^0||_p = 1$, which is easy to meet.

5.4 Stepsize Selection

Standard subgradient projection methods generally employ a constant or diminishing stepsize rule. Here, however, we can take advantage of Theorem 10 to select a more informed stepsize. Since the optimum of the ERCH is guaranteed to lie at a non–differentiable point, once an updating direction has been selected it makes sense to consider just those stepsizes that land on one of such points.

Recall from the beginning of the section that a non–differentiable point (that is, one where a non–singleton subdifferential arises) can be characterized through the orderings $W \cdot X_{i_-}^{dec}$ and $W \cdot X_{i_+}^{inc}$ as those values of W for which these orderings are not unique, i.e., some elements might be swapped without violating the ordering. In particular, only situations where equalities with the vectors $W \cdot X_{\lceil 1/\eta \rceil_\pm}$ arise can produce non–singleton subdifferentials. Therefore, we can identify non–differentiable points along the updating direction as those values of the stepsize s^t for which $W^t + s^t d^t$ produces one of such equalities, that is to say

$$(W^t + s^t d^t) \cdot X_{\lceil 1/\eta \rceil_\pm} = (W^t + s^t d^t) \cdot X_{i_\pm},$$

for some other $X_{i_\pm}$ vector in the ordering. Since several of such points can appear along the direction d^t, our approach here is to move on to the nearest of them. That is, we select the minimum stepsize (different from 0) that lands on a non–differentiable point. This approach is sensible because by moving further away we could step into a different smooth region where our current estimate of the subgradient (and thus d) is no longer valid. This results in the stepsize rule

$$s^t = \min_{i_\pm \in C_+ \bigcup C_-} \left\{ \frac{X_{\lceil 1/\eta \rceil_\pm} \cdot W^t - X_{i_\pm} \cdot W^t}{X_{i_\pm} \cdot d^t - X_{\lceil 1/\eta \rceil_\pm} \cdot d^t} \right\}, \tag{29}$$

which is obtained from solving the equality above for s^t, and taking the minimum over all of the possible equalities. The sets C_+, C_- arise from the fact that not all data points need to be checked. These sets are defined as

$$C_+ = \left\{ i \in M_+ : \begin{array}{l} X_i \cdot d^t > X_{\lceil 1/\eta \rceil_+} \cdot d^t, i < \lceil 1/\eta \rceil_+, \\ X_i \cdot d^t < X_{\lceil 1/\eta \rceil_+} \cdot d^t, i > \lceil 1/\eta \rceil_+. \end{array} \right\},$$
$$C_- = \left\{ i \in M_- : \begin{array}{l} X_i \cdot d^t < X_{\lceil 1/\eta \rceil_-} \cdot d^t, i < \lceil 1/\eta \rceil_-, \\ X_i \cdot d^t > X_{\lceil 1/\eta \rceil_-} \cdot d^t, i > \lceil 1/\eta \rceil_-. \end{array} \right\}.$$

The choice of these sets becomes clear by realizing that any point not in this set produces a negative or undefined s^t value, which is useless in our method since we are interested in advancing by following the updating direction.

We state now the following proposition, whose proof is immediate by construction of the stepsize, as presented above:

Proposition 18 RapMinos *explores a non–differentiable point at each iteration.*

Algorithm 2 RapMinos method for ERCH-NPP

Inputs: data (X, y), norm $p \in [1, \infty]$, stopping tolerance ϵ.
Initialization: chose $W^0 = W_{\eta_{min}}$, $t = 0$, $stop = \infty$.
while $stop > \epsilon$ **do**
  Find Lagrangian MNS γ_t^* solving problem (25).
  Find stepsize s^t using (29).
  Update step: $V^{t+1} = W^t - s^t \gamma_t^*$.
  Radial projection step: $W^{t+1} = R_p[V^{t+1}]$ (Eq. 28).
  Stopping criterion: $stop = ||\gamma_t^*||_\infty$.
  $t \leftarrow t + 1$.
end while
return W^t.

5.5 Initialization

While any feasible W s.t. $||W||_p = 1$ is a valid starting point, the choice of such point will determine the local minima the algorithm ends up in. As we discuss later in the experimental section, falling in a bad local minimum can result in poor classification accuracy. Therefore, it is relevant to start the optimization at a sensible W point. To do so, we propose the following heuristic. Let us consider the minimum possible value for η, which is $\eta_{min} = 1/\min\{M_+, M_-\}$. At this value each class hull gets reduced to a unique point, its barycenter, where every pattern is assigned the same weight in the convex combination. For such η, the ERCH–NPP is trivially solved by computing W as the difference between both barycenters, $W_{\eta_{min}}$. While such W will not be the solution for other values of η, intuitively we see that it will be already positioned in the general direction of the desired W_η. Although we cannot give any theoretical guarantees on such choice being a good starting point, we will see in the experimental section 6.5 how it performs well in practice.

5.6 Full Algorithm and Convergence Analysis

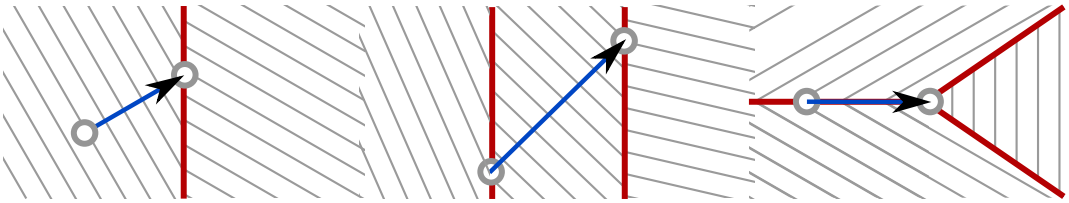
After joining the improvements presented in the previous subsections, the main steps of the full RapMinos method are presented in Algorithm 2. We show now how the iteration of such steps guarantees convergence to a local minimum of the problem. The main argument of the proof is that the RapMinos algorithm visits a region of the function at each step, but always improving the value of the objective function. Since the number of such regions is finite, the algorithm must stop at some point, having found a local minimum. The details of the proof are presented in what follows.

First we will require the following lemma:

Lemma 19 *Consider the update $W^{t+1} = R_p[W^t + s^t d^t]$ with d^t defined as in (27) and s^t defined as in (29). This update never worsens the value of the objective function, i.e., $f(W^{t+1}) \leq f(W^t)$. Furthermore, if $W^{t+1} = W^t$ then W^t is a local minimum, else $f(W^{t+1}) < f(W^t)$.*

Proof

Theorem 12 already shows that at a local minimum the update direction selected by RapMinos is null. If not at a local minimum, the updating direction is guaranteed to be



(a) (b) (c)

Figure 7: Depiction of possible scenarios arising during a RapMinos update. (a) Start in a smooth region, stop at a non–differentiable intersection between smooth regions. (b) Start at an intersection between regions, traverse a smooth region until another intersection is found. (c) Start at an intersection between regions, move along a boundary until intersection with a new smooth region is found.

a descent direction. Therefore $f(W^t + \delta d^t) \leq f(W^t)$ for some small $\delta > 0$. Consider now the structure of the objective and subgradient functions, as shown in Eqs. 16 and 20. Note that $f(W)$ is piece–wise linear, the subgradient set being a unique gradient in the interior of the linear regions, while being non–singleton in the intersections of such regions. With this in mind, the following three cases regarding the status of W^t are possible, which are also depicted in Figure 7:

- W^t is a differentiable point. Then W^t lies in a linear region, where the subgradient set is a unique constant gradient. Because of this, the Minimum Norm Subgradient of the Lagrangian is also constant throughout the whole region, and d^t remains a descent direction until a non–differentiable point marking the frontier to another region is reached (Figure 7a).

- W^t is a non–differentiable point, which means W^t is in the intersection of two or more linear regions, and $W^t + \delta d^t$ for some infinitesimal $\delta > 0$ steps in the interior of one linear region. Since the gradient in this region is included in the subgradient of W^t (see Eq. 20) and $f(W^t + \delta d^s) < f(W^t)$ is guaranteed, then moving further along this region must keep the same rate of improvement (since the region is linear), until a non–differentiable point marking the frontier to another region is reached (Figure 7b).

- W^t is a non–differentiable point and $W^t + \delta d^t$ follows an intersection of regions (e.g., follows an edge of the problem's surface). Then the MNS of the Lagrangian is not changed and d^t remains a descent direction until an intersection with a new linear region is found. This case is observed when selecting the stepsize in Eq. 29 (Figure 7c).

Whatever the case, improvement in the objective is guaranteed until the next non-differentiable point is reached. Therefore $f(W^t + s^t d^t) < f(W^t)$.

Including now the radial projection, we have that

$$\begin{aligned} f(W^{t+1}) &= f(R_p[W^t + s^t d^t]) = f\left(\frac{W^t + s^t d^t}{||W^t + s^t d^t||_p}\right), \\ &= \frac{f(W^t + s^t d^t)}{||W^t + s^t d^t||_p} < f(W^t + s^t d^t), \\ &< f(W^t), \end{aligned}$$

since $||W^t + s^t d^t||_p > 1$ (see proof for Lemma 17) and $f(cW) = cf(W)$ for c constant. ■

With this tool we are ready to prove convergence of RapMinos :

Theorem 20 *The* RapMinos *algorithm finds a local minimum in a finite number of steps.*

Proof By Proposition 18, RapMinos explores a vertex or edge of $f(W)$ at each iteration. As $f(W)$ is piece–wise linear, the number of such regions is finite, so at some point the method could step again into a previously visited point. However, this is not possible, since because of Lemma 19, each iteration must either stop at a local minimum or strictly improve the objective value, thus avoiding to return to a previous point. Therefore, RapMinos converges to a local minimum in a finite number of steps. ■

6. Experimental Results

We present now experimental results supporting our proposed ERCH model and the corresponding RapMinos algorithm, as well as details on implementation.

6.1 Implementation

The RapMinos algorithm was implemented in Matlab, and is publicly available for download [5]. The code includes an adapted RCH–NPP algorithm (Clipped–MDM, see López et al., 2011a, 2008) to solve the MNS problem (Eq. 25). The adaptation involves modifying the algorithm to accept the sets of points $\mathcal{Q}_\pm$, which must always retain a coefficient $\mu_i = \eta$ and thus are not optimized over, but nevertheless should be taken into account when computing the objective value. This can be done easily by adapting the initialization and extreme points computation at the end of the algorithm: for further details please refer to the code itself.

A point of technical difficulty in the implementation is the bookkeeping of the index sets $\mathcal{Q}_\pm$, $\mathcal{S}_\pm$. While these could be recomputed from scratch each time they are needed, it is far more efficient to update them throughout the iterations. To do so, at the initialization of RapMinos these sets are built using the initial vector W^0. After that, during the algorithm iterations, these sets are updated at two situations:

- When computing the stepsize using (29), the pattern (or patterns) that produce the min are added to their respective $\mathcal{S}_\pm$ set. This is done because, by definition of the

5. Project web page: `https://bitbucket.org/albarji/rapminos` . Source code and packages available.

stepsize rule, the margin of this pattern after the update equals that of $W \cdot X_{\lceil 1/\eta \rceil}$, and this is what defines the $\mathcal{S}_{\pm}$. This pattern is also removed from the set $\mathcal{Q}_{\pm}$ in the case it was part of it.

- After each MNS computation the values of the weights μ_i for the patterns in the sets $\mathcal{S}_{\pm}$ are checked. If any of them turns out to be 0, it is removed from $\mathcal{S}_{\pm}$, since such pattern has no longer an influence in the subgradient. If it happens to be valued η, then the pattern is transferred to the corresponding $\mathcal{Q}_{\pm}$ set.

Because of numerical errors amounting during the algorithm iterations, such checks are always done with a certain tolerance value. Also, for the same reason, it could happen that an update of the algorithm worsens the value of the objective function, even if this is theoretically impossible thanks to Lemma 19. To address this, our implementation stops whenever a worsening is detected.

Regarding the quality of the solution obtained, it should be noted that the RapMinos algorithm solves the intersecting ERCH–NPP case, and most of its assumptions are based on this fact. To avoid convergence problems if the problem is actually non–intersecting, our implementation first invokes a standard RCH–NPP solver. If the solution W obtained has norm close to zero, the problem might be intersecting. To check whether there is a real intersection we solve the following linear program

$$\begin{aligned} \min_{\lambda,\eta} \quad & \eta, \\ \text{s.t.} \quad & \begin{cases} \sum_{i\in M_+} \lambda_i X_i = \sum_{i\in M_-} \lambda_i X_i, \\ \sum_{i\in M_+} \lambda_i = 1, \sum_{i\in M_-} \lambda_i = 1, \\ 0 \le \lambda_i \le \eta, \ \forall\, i. \end{cases} \end{aligned} \tag{30}$$

which finds the minimum value of η for which the reduced convex hulls intersect. If the user–selected value of η is larger than the one found here, then the hulls intersect, and we continue with the execution of RapMinos . Otherwise, a solution is obtained by solving the equivalent ℓ_p RCH–NPP (Eq. 11) through a generalized RCH–NPP solver; details on this solver are outlined in the Appendix.

6.2 Augmented Model Capacity: Synthetic Data Sets

We first show how the augmented ν range extension of the Eν–SVM model, and thus ERCH–NPP, can improve the classification accuracy of the SVM. As shown in Section 3, the ERCH–NPP model is able to generate non–trivial solutions for those cases where the reduced hulls of the data intersect, on top of all the solutions attainable by the standard RCH–NPP model for non–intersecting hulls. We hypothesize that this capability ought to be specially useful in classification problems where the convex hulls of positive and negative classes have a significant intersecting area, as RCH–NPP would only be able to find useful solutions for a small range of η values. A similar hypothesis was previously proven for other margin based methods when replacing the regularization constraint $\|W\|_p \le 1$ by $\|W\|_p = 1$ or replacing the reduced convex hulls $\mathcal{U}_{\pm}$ by different class shapes (for example, ellipsoids), as shown in Takeda et al. (2013).

To test this, we generated a series of artificial data sets with increasingly larger intersecting areas. We defined conditional probabilities for label +1 and label -1, denoted by $p(X|+1)$ and $p(X|-1)$, as multivariate normal distributions. The mean vector and the variance–covariance matrix of $p(X|+1)$ were defined by the null vector $(0,\ldots,0)^\top \in \mathbb{R}^n$ and the identity matrix $\mathbf{I}_n \in \mathbb{R}^{n\times n}$, respectively (i.e., standard normal distribution). For the other conditional probability, $p(X|-1)$, we randomly generated the variance-covariance matrix having eigenvalues $0.1^2,\ldots,1.5^2$, wherein the square roots of the eigenvalues were numbers placed at even intervals from 0.1 to 1.5. The mean vector of $p(X|-1)$ was defined by $\frac{r}{\sqrt{n}}(1,\ldots,1)^\top \in \mathbb{R}^n$, with r a distance parameter between classes. The larger the r, the smaller the intersecting area between classes. The training sample size and test sample size were set to $m = 2\times 10^3$ and $\widetilde{m} = 10^4$, respectively, while the number of features was chosen as $n = 10$.

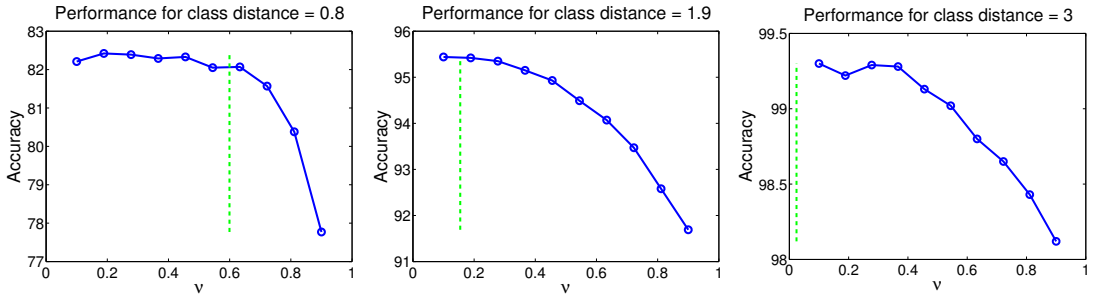


Figure 8: Performance of the Eν–SVM model for a classification problem with different degrees of distance between class centers. For each distance choice, accuracy of the trained classifier is shown for a range of ν values. The green dashed lines represent the ν threshold below which the reduced hulls intersect, hence producing a non–convex problem.

Figure 8 shows the obtained accuracy levels with RapMinos for the range $\nu \in [0.1, 0.9]$ and a selection of class distances. The threshold for which the reduced–convex–hulls intersect is also shown, below which the problem becomes non–convex and only the ERCH–NPP model can find meaningful solutions. As expected, when the distance between class means is large, this threshold becomes smaller, as a smaller ν implies a larger η, i.e., a smaller reduction on the convex hulls is required for them to become separable. For those cases where the distance between classes is small, the intersecting range of ν shows an improvement on accuracy over the non–intersecting range, thus backing up the fact that the augmented range of ERCH–NPP (and so Eν–SVM) can lead to more accurate models.

6.3 Augmented Model Capacity: Real–World Data Sets

We now test the benefits provided by the augmented model capacity on real–world data sets, obtained from the benchmark repository at Rätsch (2000), but instead of making use of the default 100 training–test partitions provided there we generated our own random splits of each data set as done in Takeda and Sugiyama (2009). In particular, we took 4/5 of the data set as training data and the remaining 1/5 as testing data. For each data set

we identified the ν_{limit} value for which the class hulls start intersecting, and solved ERCH–NPP for two ranges of ν values of 100 points each, one above ν_{limit} (convex range), and the other below it (non–convex range). To solve the ERCH–NPP in the non–convex range we resorted to the presented RapMinos method, while for the convex range we applied the standard ν–SVM solver provided in LIBSVM (Chang and Lin, 2001).

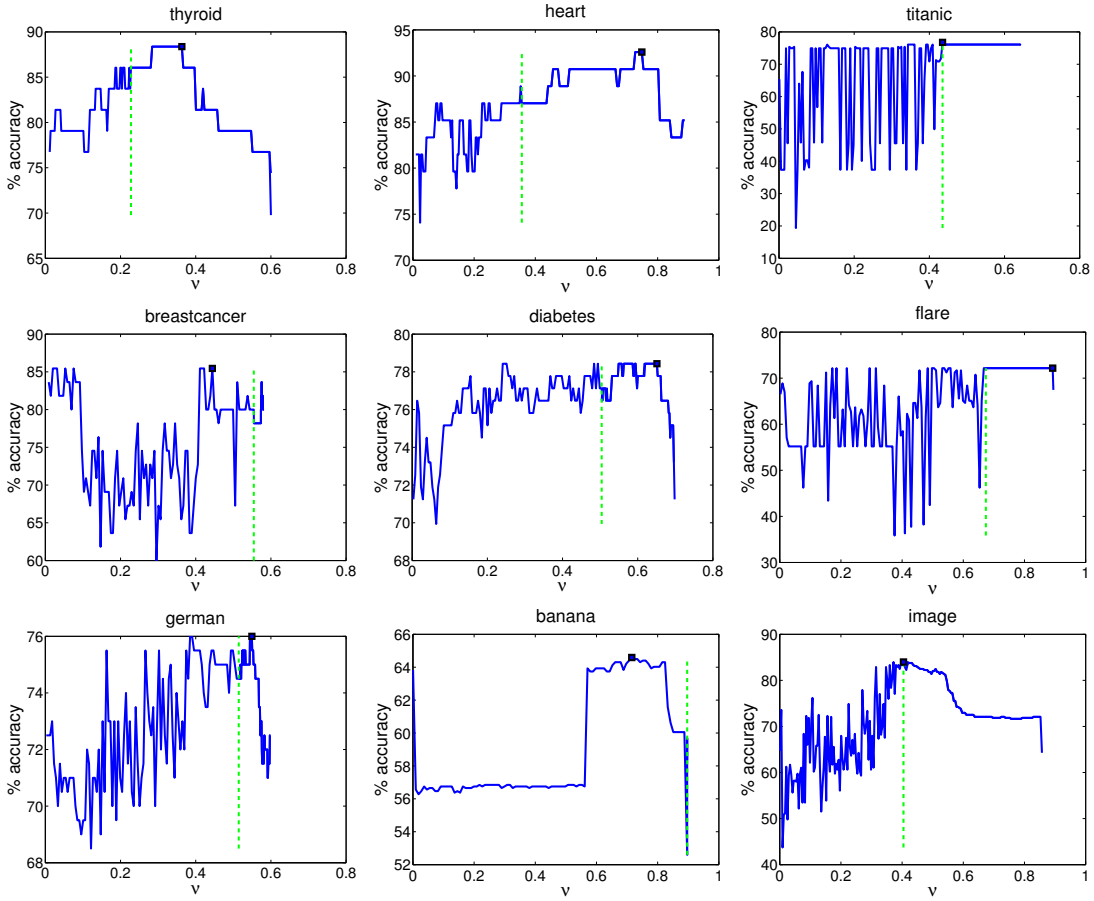


Figure 9: Performance of the Eν–SVM model for a set of real–world data sets. The square markers denote the best performing ν choice.

Figure 9 shows the accuracy levels obtained with RapMinos for the full range of ν values. While for a number of the data sets the augmented ν range does not provide noticeable benefits, for *titanic*, *breastcancer*, *ringnorm* and specially *banana* higher levels of accuracy are attainable.

Table 1 presents top accuracy values in the whole ν range for the standard ν–SVM and the augmented Eν–SVM model tuned with different ℓ_p–norm choices. The results seem to confirm our hypothesis stating that the ability to select an arbitrary ℓ_p regularization in the model leads to an increase in the model capacity: in 8 out of 13 data sets we find that the model is able to obtain higher accuracy values than both the ν–SVM and ℓ_2 Eν–SVM

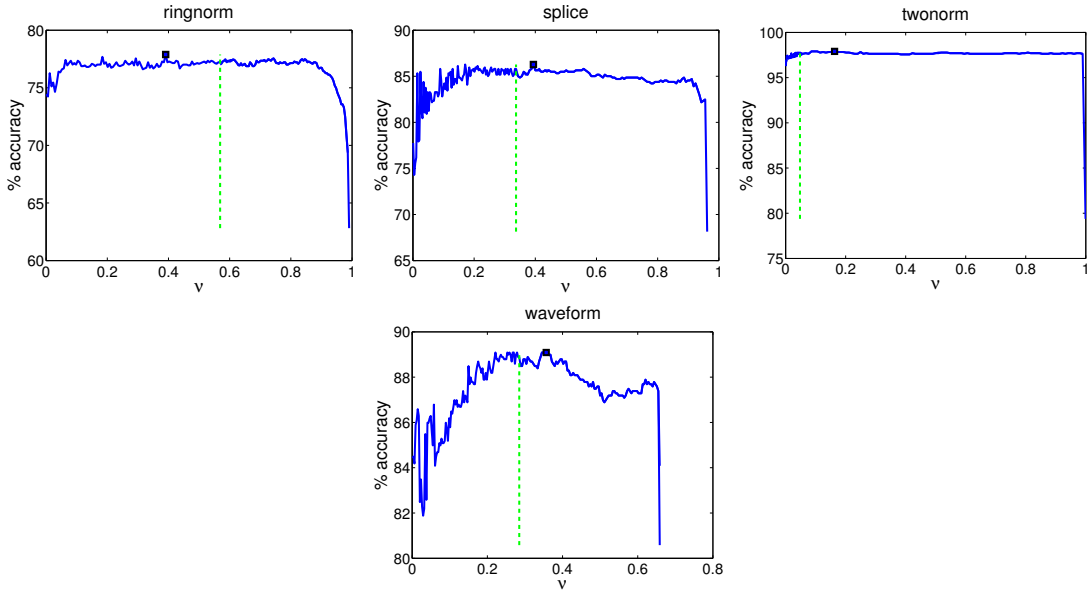


Figure 9: (continued) Performance of the Eν–SVM model for a set of real–world data sets. The square markers denote the best performing ν choice.

models. For illustration purposes we also include the accuracy curves for a sample of the data sets in Figure 10.

6.4 Runtime Experiments

To show the advantage in terms of efficiency and stability of the proposed RapMinos algorithm we present here a comparison against a reference Eν–SVM method. Recall the Eν–SVM problem is dual to the ERCH–NPP discussed here (see Proposition 2), so in principle similar solutions should be obtained through both approaches, although it should be noted that the existence of local minima in both models can lead to different results. The method of choice for the Eν–SVM problem is the one presented in Takeda and Sugiyama (2008) [6], which finds a solution by approximating the non–linear Eν–SVM problem by a series of linear optimization problems; such linear problems, in turn, are solved by invoking an interior–point method.

We worked again with the data sets from the benchmark repository at Rätsch (2000), but since we wanted to test the algorithms in the intersecting range of data, instead of selecting ν as the value maximizing validation accuracy we fixed it at a value slightly below the separable limit ν_{min}. Table 2 shows training times for the reference Eν–SVM and the RapMinos algorithms, together with the accuracy levels obtained in the test splits. A basic subgradient projection method solving ERCH–NPP (see Algorithm 1) is also included in the table to check whether the theoretical improvements provided by RapMinos have noticeable effects in practice.

6. This method turns out to be a subtle modification of the original Eν–SVM method by Pérez-Cruz et al. (2003).

Data set	ν–SVM	ERCH–RapMinos				
	ℓ_2	ℓ_2	ℓ_1	$\ell_{1.5}$	ℓ_3	ℓ_∞
thyroid	88.4%	88.4%	86.0%	88.4%	**95.3%**	90.7%
heart	92.6%	92.6%	90.7%	**94.4%**	92.6%	92.6%
titanic	76.1%	**76.8%**	76.1%	76.8%	76.8%	**77.2%**
breastcancer	83.6%	**85.5%**	81.8%	83.6%	83.6%	83.6%
diabetes	78.4%	78.4%	78.4%	**79.1%**	78.4%	78.4%
flare	72.2%	72.2%	72.2%	72.2%	72.2%	70.3%
german	76.0%	76.0%	76.5%	76.0%	76.0%	**77.0%**
banana	53.2%	**64.6%**	64.2%	64.5%	61.1%	64.6%
image	84.0%	84.0%	71.2%	81.8%	79.4%	78.4%
ringnorm	77.6%	**77.9%**	77.8%	77.7%	**78.0%**	**78.0%**
splice	86.3%	86.3%	86.0%	86.3%	85.8%	85.8%
twonorm	97.9%	97.9%	97.9%	97.9%	97.8%	**98.0%**
waveform	89.1%	89.1%	89.2%	**89.3%**	**89.3%**	89.1%

Table 1: Test accuracies for ν–SVM and the ERCH model trained with RapMinos , for different values of the ℓ_p–norm. Numbers in bold in the RapMinos ℓ_2 mark when the ERCH model performs better than the standard ν–SVM. Also marked in bold are those cases where a non–standard ℓ_p norm produces further improvement.

The first thing to observe is that the Eν–SVM algorithm used failed to produce a solution for some of the data sets. These failures stem from instability issues of the interior–point solver, which at some situations was unable to find a suitable interior point. Opposite to this, RapMinos always found a solution. Not only that, but also did so in considerably less time and with a higher degree of accuracy in the solution. This last fact can be explained by realizing that while the Eν–SVM approach finds a solution by using a series of linear approximations to the non–convex Eν–SVM problem, RapMinos instead addresses the non–convex ERCH–NPP problem directly. As a whole, RapMinos is able to find better–quality solutions consistently at a lower computational cost.

Regarding the improvements of RapMinos over a basic subgradient projection method, Table 2 shows how RapMinos was able to find a solution much faster for most of the data sets. Some notable exceptions are *breastcancer*, *diabetes* and *flare*, where the simple subgradient method finds a good solution quite fast. Table 3 reveals additional insight into this: the solutions found by RapMinos tend to produce better objective values. Which is to say, subgradient projection might return a solution faster in some settings, but performs a worse optimization job. It is thus clear that RapMinos is a better solver for the ERCH–NPP problem than a basic subgradient projection method, as we hypothesized when we proposed the method.

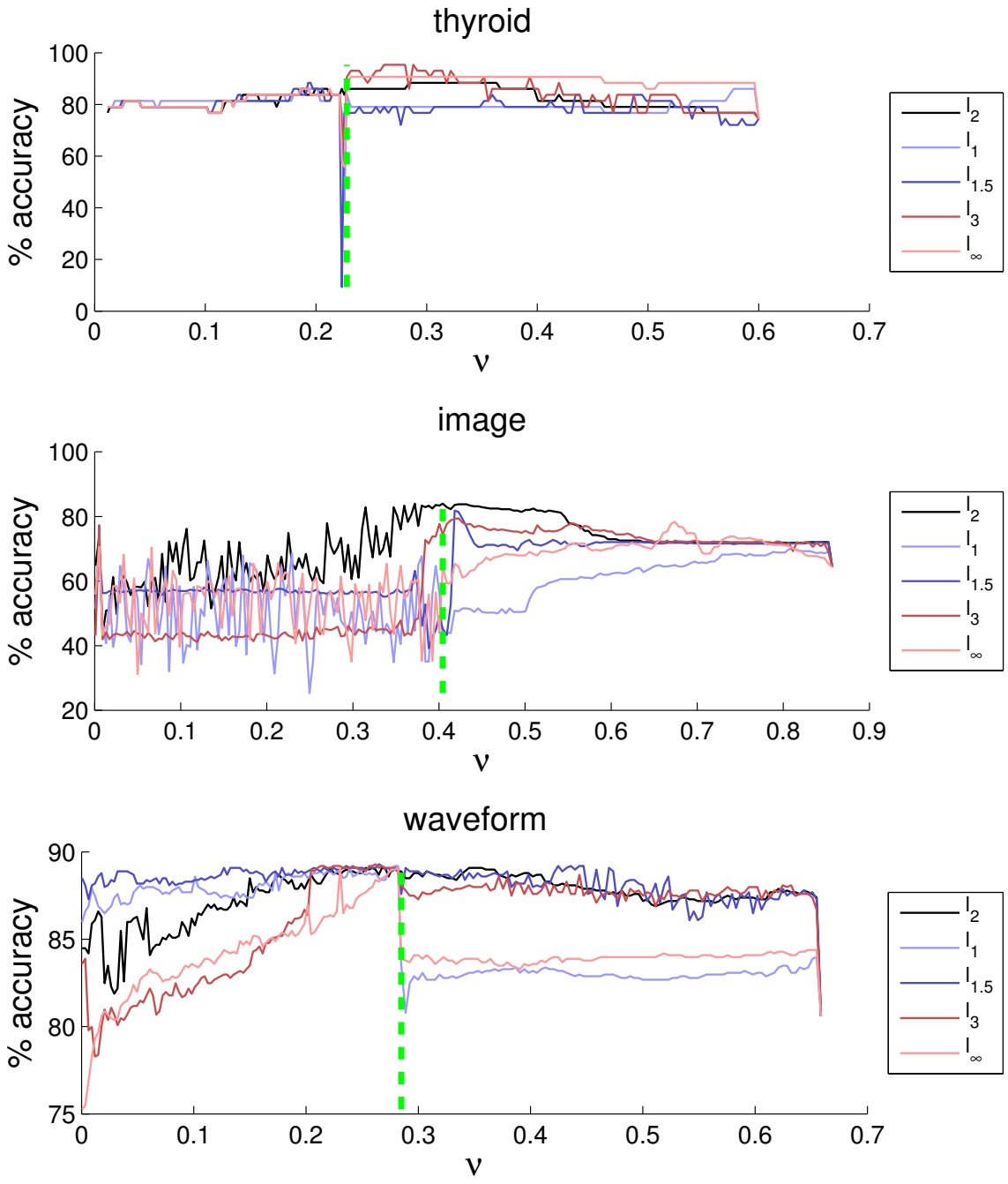


Figure 10: Performance of the Eν–SVM model for a set of real–world data sets and different values of the ℓ_p–norm.

Data set	Eν–SVM solver		Subgrad. Proj.		RapMinos	
	Accuracy	Time	Accuracy	Time	Accuracy	Time
thyroid	80.8%	1.46	86.3%	20.29	86.3%	0.15
heart	82.6%	1.72	73.9%	21.46	73.9%	0.37
titanic	76.5%	1.80	77.82%	29.84	72.9%	0.30
breastcancer	78.7%	1.05	76.6%	0.23	72.3%	0.35
diabetes	73.5%	3.21	75.1%	0.10	74.7%	0.47
flare	–	–	63.0%	0.01	63.3%	0.20
german	66.56%	2.98	77.3%	1.48	77.3%	1.33
banana	–	–	60.5%	44.96	60.5%	0.53
image	–	–	82.1%	35.89	75%	1.15
ringnorm	77.1%	26.85	77.1%	81.95	77.1%	7.24
splice	51.9%	33.27	83.7%	46.42	84.2%	9.76
twonorm	97.7%	24.05	97.2%	61.33	97.2%	11.02
waveform	78.8%	14.85	86.9%	54.21	86.9%	7.94

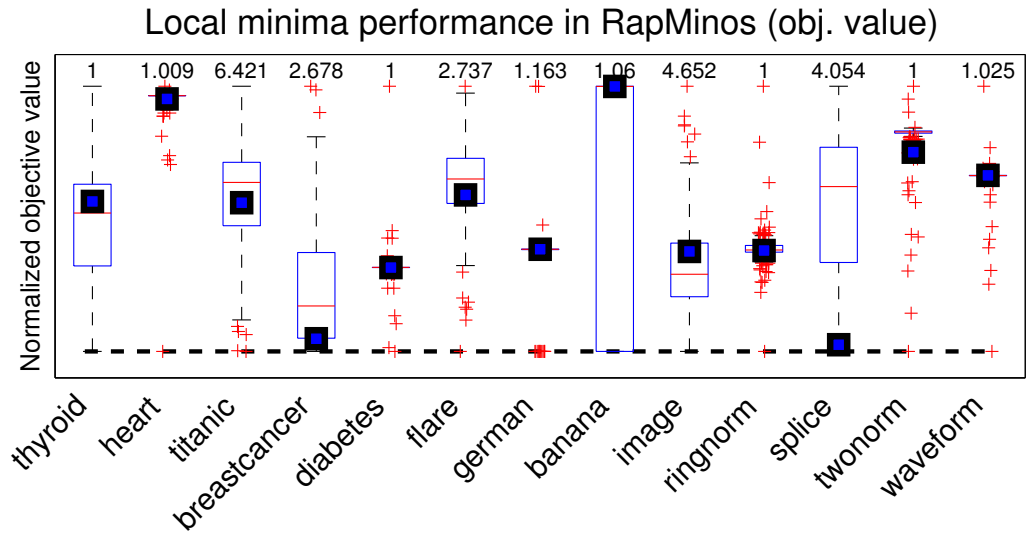
Table 2: Execution times (in seconds) and accuracy in the test set for the reference Eν–SVM solver, the proposed RapMinos algorithm and a simple subgradient projection method. Entries marked with – stand for executions where the Eν–SVM solver failed to produce a solution at all.

6.5 Quality of Local Minima

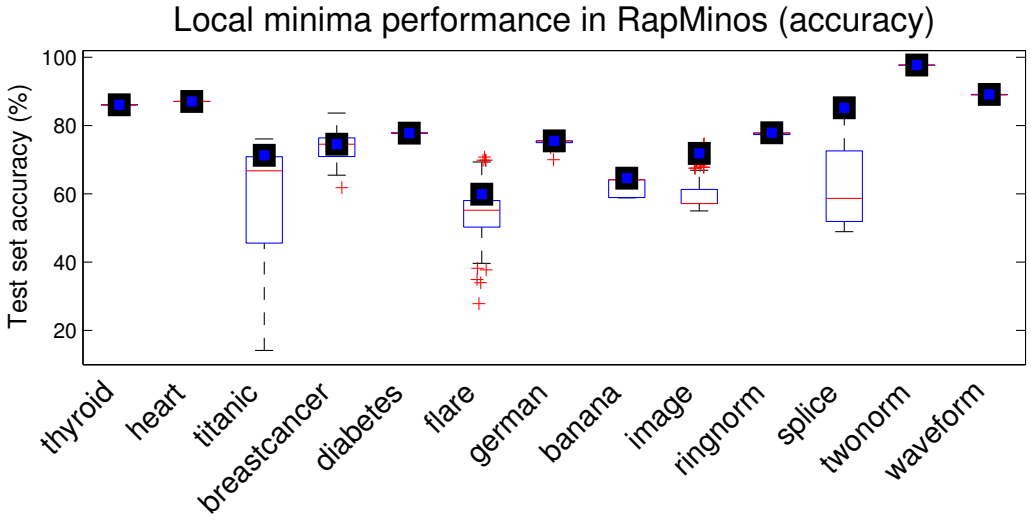
Since in the intersecting case of ERCH–NPP the optimization problem becomes non–convex (see Section 4), RapMinos only finds a local minimum of the problem. Such local minimum might or might not have an objective value similar to the overall global minimum of the problem, and so it might be the case that RapMinos finds a "bad local minimum" where a poor solution is obtained. This kind of problem is quite similar to the issues appearing in multilayer neural network training (Duda et al., 2001), where the non–linearity of the model allows to find only locally optimal solutions. Although several approaches have been proposed to address this issue, the most effective ones involve heuristics for model weights initialization that, while not guaranteeing global optimality, provide some practical means to avoid bad local minima.

In section 5.5 we proposed a heuristic to select the starting point for RapMinos . We will show now that such initialization strategy proves to be helpful in avoiding local minima. For doing so, for each data set in section 6.3 we ran the RapMinos algorithm using the presented approach, and compared the value of the objective function (Equation 7) against 200 runs with random starting points. We fixed $p = 2$ and chose ν as the one giving the highest validation performance, and for those data sets where ν was in the separable range, we chose a ν value slightly below the one for which hulls start intersecting. This way all tests were run for the intersecting case.

Figure 11a presents box plots on the distribution of such objective value for all data sets, comparing also against the value obtained with the proposed initialization. Being



(a)



(b)

Figure 11: Distribution of a) objective values (lower is better) and b) accuracies (higher is better), obtained by RAPMINOS for several data sets. The box plots represent the distribution of objective values and accuracies for the runs with random initialization, and the square markers the value obtained when using the proposed initialization heuristic. Objective values are normalized to present the best minimum found at the bottom line, while the worst one is shown at the top along with a multiplier representing how far away it is from the best value (worst = multiplier · best). A multiplier value of 1 is shown when the best and worst values are equal down to the fourth significant digit.

Data set	RapMinos	Subgrad. Proj.
thyroid	0.179	0.233
heart	0.589	0.586
titanic	-0.808	2.283
breastcancer	1.194	1.411
diabetes	0.679	0.748
flare	5e-09	-0.001
german	-9e-07	0.053
banana	1.072	1.109
image	-4e-06	0.011
ringnorm	0.080	0.099
splice	1.305	1.287
twonorm	0.755	0.755
waveform	1.783	1.759

Table 3: Objective values after optimization in RapMinos and a simple subgradient projection method. Lower is better.

a heuristic procedure, our proposal does not guarantee good local minima in all cases, though nevertheless finds solutions closer to the overall best minimum more frequently than employing a random initialization. Figure 11b presents analogous results when measuring accuracy on the test set, where again a random initialization performs worse than our proposed heuristic initialization.

7. Conclusions and Further Work

In this work we have given a geometrical interpretation of the Eν–SVM formulation, establishing connections from this model to other well–known models in the SVM family. Not surprisingly, while Eν–SVM generalizes ν–SVM to cover the case where ν is too small, this new interpretation generalizes the usual geometric viewpoint of ν–SVM finding the nearest points of two non–intersecting reduced convex hulls (RCH–NPP). Specifically, it also allows these reduced–convex–hulls to intersect, that is, it also covers the case where the reduction η coefficient is too large.

We have also proposed the RapMinos method and shown how it is able to solve the ERCH–NPP problem efficiently and for any choice of $\ell_{p\geq 1}$–norm. This not only allows to build Eν–SVM models faster than with previously available methods, but also provides even more modeling capabilities to the SVM through the flexibility to work with these different norms.

From the light of the experiments, it would seem that the Eν–SVM model can improve classification accuracy for those problems where there is a significant intersection between class hulls. The added ℓ_p–norm flexibility has also proven to be useful to increase classification accuracy in a number of data sets, extending further the applicability of the model.

A number of interesting extensions to this work, which would require further research efforts, are possible. While the RapMinos method finds a solution efficiently and we provide some empirical evidence on it being a reasonably good local minimum, the method is still far from finding global minima. Even though finding global minimizers for non–convex problems is a daunting challenge, a globalization strategy based on concavity cuts has already been developed for the Eν–SVM model (Takeda and Sugiyama, 2008). Whether this approach is also applicable to the dual ERCH–NPP problem is an open issue. Finally, in this paper we have only addressed linear models. Extending the methods here to address kernelized models is also an open problem.

Acknowledgments

This work has been partially supported by Spain's TIN 2010–21575–C02–01 and TIN 2013–42351–P projects and by Cátedra UAM–IIC en Modelado y Predicción.

Appendix A. Proof for Theorem 10 (Optimum at Non–Differentiable Points)

Consider the Minkowski polygon representation of the ERCH-NPP (Eq. 17). If the constraint $||W||_p = 1$ is ignored, the problem would become

$$\min_{W} \quad \max_{X \in \mathcal{M}} \quad W \cdot X.$$

This problem clearly involves the minimization of a piece–wise linear function, where the pieces are determined by the inner maximization $\max_{X \in \mathcal{M}} \quad W \cdot X$. Consider now one of such pieces, which we shall denote S. For every $W \in S$ the inner maximization problem selects the same solution X_S, and so the minimization in this piece can be written as

$$\min_{W \in S} \quad W \cdot X_S.$$

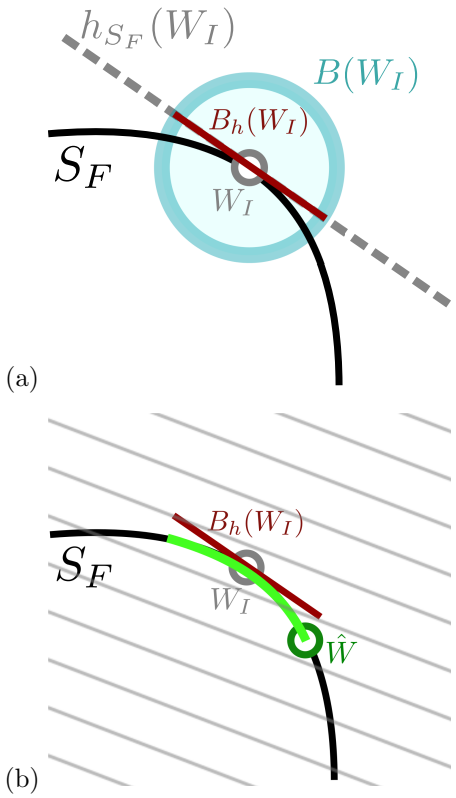
Since this is a linear problem, the optimum necessarily lies at a boundary point of S, that is, at the frontier with another linear region of the global problem, this frontier being a non–differentiable region. However, when taking the constraint back into account we have

$$\min_{W \in S, ||W||_p = 1} \quad W \cdot X_S.$$

which is no longer a linear problem, since the norm constraint on W defines a non–convex feasible set. Hence, the minimum in this linear region need not lie at an extreme. Nevertheless, we show in what follows that this property is still met regardless of this constraint.

Let us denote S_F as the feasible region within S, that is, $S_F \equiv \{W | W \in S, ||W||_p = 1\}$. This region is a surface which is a subset of the ℓ_p unit–ball. To show that the minimum in this region always lies at an extreme point we will assume that, on the contrary, the optimum is in a non–extreme point W_I. We will then see that there always exists another point in a neighborhood of W_I presenting a better or equal value of the objective function.

Consider the supporting hyperplane of S_F at W_I, $h_{S_F}(W_I)$ (see Definition 13). This hyperplane can always be defined for any interior point of S_F as the hyperplane tangent to S_F at W_I. This hyperplane leaves all of S_F at one side. Consider also a ball $B(W_I)$ of small radius $r > 0$, centered on W_I, which shall be understood as a neighborhood of W_I. Let us define $B_h(W_I)$ as the intersection of this ball and the supporting hyperplane, $B_h(W_I) \equiv B(W_I) \bigcap h_{S_F}(W_I)$. This set does define a convex set, since it is the intersection of a hyperplane and a sphere. Because of that, the objective function $W \cdot X_S$ for $W \in B_h(W_I)$ always has a minimizer at an extreme of the set. More precisely, $\exists\, v^* \in B_h(W_I), v^* \neq W_I$ so that $v^* \cdot X_S \leq W_I \cdot X_S$. Thus, there exists a small displacement along a support hyperplane from a non–extreme point W_I that cannot worsen the value of the objective function. But of course, v^* might not be a feasible point, since by the properties of the supporting hyperplane all the points $v \in B_h(W_I)$ have $||v||_p \geq 1$.



(c) Visual example of the concepts introduced for the proof of Theorem 10. a shows the feasible region within a linear region of the problem (S_F), the supporting hyperplane at an interior point of this region ($h_{S_F}(W_I)$), the ball defining the neighborhood ($B(W_I)$) and its intersection with the supporting hyperplane ($B_h(W_I)$). b shows how this intersection can be projected back to the feasible region S_F, and how an extreme of it is able to obtain a better value of the objective function (represented through its level sets as gray lines).

The next step is showing that projection of v^* back to the feasible region S_F still guarantees that the projected point cannot be worse than the initial W_I in terms of the value of the objective function. First, it must be realized that the radial projection $R_p[v]$ for

any $v \in B_h(W_I)$ always results in feasible points inside S_F. This is immediate by realizing that the radial projection just rescales the norm of its vector argument, and so

$$\arg\min_{X\in\mathcal{M}} R_p\left[v\right]\cdot X = \arg\min_{X\in\mathcal{M}} \frac{v\cdot X}{||v||_p} \equiv \arg\min_{X\in\mathcal{M}} v\cdot X,$$

i.e., the solution of the internal problem does not change, and so the projected v remains in the same linear region S. Using then the properties of the radial projection, $||R_p\left[v\right]||_p = 1$, and so $R_p\left[v\right] \in S_F$.

Now note that since we also have that $\forall\ v \in B_h(W_I)$, $||v||_p \geq 1$, then the radially projected points can be defined as $R_p\left[v\right] = \frac{v}{||v||_p} = c(v)\ v$, for some scalar $c(v) \in (0,1]$. Also, since $W\cdot X_S \geq 0$, $\forall\ W \in S_F$ (because of the intersecting hulls, see Lemma 7), we can establish the following chain of relationships

$$\begin{aligned}\min_{v\in B_h(W_I)} R_p\left[v\right]\cdot X_S &= \min_{v\in B_h(W_I)} c(v)\ v\cdot X_S \\ &\leq \min_{v\in B_h(W_I)} v\cdot X_S \\ &\leq W_I\cdot X_S.\end{aligned}$$

Therefore, any non–extreme point $W_I \in S_F$ has always a feasible neighbor which presents an equal or better value of the objective function, and so W_I cannot be optimal (or at least there exists another point with an equally optimal value). Extending this argument to every non–extreme point in S_F, we can conclude that there exists an extreme point W_E such that $W_E\cdot X_S \leq W\cdot X_S$, $\forall\ W \in S_F$. Consequently, a minimizer of the global problem always lies at the intersection between two linear regions, that is to say, at a non–differentiable point. ■

Appendix B. Proof for Theorem 12 (Descent Directions for ERCH)

To prove this theorem we need to resort to some tools from the field of non–convex non–smooth analysis, most of them contained in Clarke (1990). Nevertheless, for completeness of the paper we will briefly introduce such required tools here.

Consider a general constrained optimization problem in the form

$$\begin{aligned}\min_{x\in X}\quad & f(x),\\ \text{s.t.}\quad & g_i(x)\leq 0,\quad i=1,\ldots,n,\end{aligned}$$

where any equality constraint in the form $h(x) = 0$ can also be taken into account by producing two inequality constraints $h(x) \leq 0$, $h(x) \geq 0$.

We introduce now the concept of relative subdifferential as

Definition 21 *Relative subdifferential: given the set $S \subseteq X$, the S–relative subdifferential of f at x, $\partial|_S f(x)$ is defined as*

$$\partial|_S f(x) = \left\{\xi\,|\xi_i\to\xi, \xi_i\in\partial f(y_i), y_i\in S, y_i\to x\right\},$$

that is to say, it is the set of subgradients appearing when approaching x from a succession of points y_i tending to x. In the event that $x \notin S$, $\partial|_S f(x) = \emptyset$.

Consider now the augmented objective function

$$F(x) = \max\left\{f(x) - f(x^*), g_1(x), \ldots, g_n(x)\right\},$$

where $f(x^*)$ is the optimal value of the original objective function. Observe that at the optimum of the original problem, $F(x^*) = 0$, since all constraints are met ($g_i(x) \leq 0$) and the first term takes the value 0. Let us define the set

$$\Gamma(x) = \text{conv}\left\{\partial f(x), \partial|_{G_1(x)} f(x), \ldots, \partial|_{G_n(x)} f(x)\right\},$$

where $G_i(x)$ is the set of points for which the constraint $g_i(x)$ is not feasible ($g_i(x) > 0$). $\Gamma(x)$ can be interpreted as a kind of subdifferential of the Lagrangian. We then have two results associated with this set (Clarke, 1990, Theorem 6.2.2. and Proposition 6.2.4.):

- If x is a local minimum of the problem, then $0 \in \Gamma(x)$.
- Else, let γ be the element of $\Gamma(x)$ with minimum norm. Then $d = -\gamma$ is a descent direction in $F(x)$.

In other words, if we are not already at the optimum, performing a small step in the direction of d reduces the value of the augmented function $F(x)$. Note that, given the form of $F(x)$, this guarantees that either the objective function $f(x)$ or the violation in some constraint is reduced.

Let us apply now these tools to the ERCH problem $\min_W f(W)$ s.t. $||W||_p = 1$. The augmented function $F(W)$ comes easily as

$$F(W) = \max\left\{f(W) - f(W^*), ||W||_p - 1, 1 - ||W||_p\right\},$$

where the equality constraint has been rewritten as two inequalities. Now, taking into account the fact that in our algorithm we guarantee $||W||_p = 1$ at every iteration, the max in $F(W)$ is always attained for the first term when not at the optimum. Also because of this we have that $\partial|_{G_1}||W||_p = \partial|_{(||W||_p > 1)}||W||_p = \partial||W||_p$, and similarly for $\partial|_{G_2}||W||_p$. That is to say, the relative subdifferential coincides with the standard one. Therefore, the set $\Gamma(W)$ results to be

$$\Gamma(W) = \text{conv}\left\{\partial f(W), \partial||W||_p, -\partial||W||_p\right\}.$$

We can rewrite this set in a more convenient form as

$$\begin{aligned}\Gamma(W) &= \mu_1 \partial f(W) + \mu_2 \partial||W||_p - \mu_3 \partial||W||_p, \\ &= \mu_1 \partial f(W) + (\mu_2 - \mu_3)\partial||W||_p,\end{aligned}$$

where the convex coefficients meet the usual constraints $\sum_i \mu_i = 1$, $0 \leq \mu_i \leq 1$. It should be realized now that the gradient of the norm $\partial||W||_p$ is the 0 vector only at the origin $W = 0$,

which is an infeasible point. Therefore, at the optimal W^* it will be necessary to combine this gradient with $\partial f(W)$ to produce the 0 vector bound to appear at a local minimum in $\Gamma(W)$, and so the coefficient must be non–zero, $\mu_1 > 0$. We can then divide the expression by μ_1 [7], obtaining

$$\begin{aligned}\Gamma(W) &\equiv \partial f(W) + \frac{\mu_2 - \mu_3}{\mu_1}\partial||W||_p, \\ &= \partial f(W) + \lambda\partial||W||_p,\end{aligned}$$

for $\lambda = \frac{\mu_2-\mu_3}{\mu_1} \in \mathbb{R}$. It is realized now that the expression obtained for $\Gamma(W)$ is actually the standard subdifferential of the Lagrangian.

Invoking now the properties of the set $\Gamma(x)$ stated above, it is immediate that at local minimum $\arg\min_W ||\Gamma(W)|| = 0$. Descent in the original function $f(x)$ is also obtained by realizing that the direction $d = -\arg\min_W ||\Gamma(W)||$ guarantees descent in $F(W)$, and so at a point $W' = W + sd$, with $s > 0$ sufficiently small,

$$\begin{aligned}f(W') - f(W*) &< \max\left\{f(W') - f(W^*), ||W'||_p - 1,\right. \\ &\qquad \left. 1 - ||W'||_p\right\}, \\ &= F(W') < F(W), \\ &= \max\left\{f(W) - f(W^*), ||W||_p - 1,\right. \\ &\qquad \left. 1 - ||W||_p\right\}, \\ &= f(W) - f(W^*),\end{aligned}$$

since at W the constraints are met. Therefore $f(W') < f(W)$, and so d is also a descent direction for $f(W)$, concluding the proof. ■

Appendix C. Computation of the Derivative of the Constraint

Depending on the actual value of the norm parameter $p \geq 1$, the norm function $||W||_p^p$ might produce a singleton or a set of subgradients. For even p the norm function is smooth an thus produces a singleton subgradient in the form

$$\begin{aligned}\left[\frac{\partial||W||_p^p}{\partial W}\right]_k &= \frac{\partial}{\partial W_k}\sum_i (W_i)^p, \\ &= p\,(W_k)^{p-1}.\end{aligned}$$

However, for an odd or non–integer value of p the absolute value function cannot be disposed of, and the set of subgradients produced takes the form

7. Even though this transformation changes the scaling of the points in the set $\Gamma(W)$, note that the argument remains legit, since we are only interested in extracting a direction vector from $\Gamma(x)$, and therefore scaling is not relevant.

$$\begin{aligned}\left[\frac{\partial||W||_p^p}{\partial W}\right]_k &= \frac{\partial}{\partial W_k}\sum_i |W_i|^p\,,\\ &= p\,|W_k|^{p-1}\,\frac{\partial}{\partial W_k}|W_k|,\\ &= p\,|W_k|^{p-1}\,\mu_k,\end{aligned}$$

where the coefficients μ_k take the values

$$\mu_k = \begin{cases} 1 & \text{if} \quad W_k > 0,\\ -1 & \text{if} \quad W_k < 0,\\ [-1,1] & \text{if} \quad W_k = 0.\end{cases}$$

That is to say, for values of W with entries at 0 several possible subgradients appear. Nevertheless, since if $W_k = 0$ then $|W_k|^p = 0$ (except for $p = 1$, see below), the particular choice of μ_k is irrelevant, and we end up at

$$\left[\frac{\partial||W||_p^p}{\partial W}\right]_k = p\,|W_k|^{p-1}\,\mathrm{sign}(W_k).$$

as shown in Eq. (23).

The cases $p = 1$ and $p = \infty$, which are of special relevance for their known sparsity/uniformity inducing properties, require some further attention. First, for $p = 1$ we have

$$\left[\frac{\partial||W||_1}{\partial W}\right]_k = \frac{\partial}{\partial W_k}\sum_i |W_i| = \mu_k,$$

and a similar situation to that of the general p arises, though this time the particular choice of μ_k does produce different subgradients. This is not surprising, since the ℓ_1–norm is non–smooth. To address this issue, in this paper we take the simplest of the available subgradients, taking $\mu_k = 0$ whenever $W_k = 0$, resulting in

$$\left[\frac{\partial||W||_1}{\partial W}\right]_k = \mathrm{sign}(W_k).$$

It must be noted, however, that by making this simplification we might be failing to identify the correct updating directions in our algorithm when standing on a W point where the norm is not differentiable. This, however, poses no problems to our method in practice, but for very specifically tailored cases unlikely to arise in practice. Even in those cases the solution of the ERCH with norm ℓ_1 can be safely approximated by a norm choice like $\ell_{1.001}$, which is smooth.

Now for $p = \infty$ the derivative is, in principle, not separable, since we have

$$\frac{\partial||W||_\infty}{\partial W} = \frac{\partial}{\partial W}\max\{|W_i|\}\,.$$

Nevertheless we can rewrite this as

$$\frac{\partial||W||_\infty}{\partial W} = \frac{\partial}{\partial W}\max\{W_1, -W_1, \dots W_n, -W_n\},$$

and invoke again the property that the subdifferential of the maximum of a set of convex functions (linear, in this case) at a given point is the convex hull of the subdifferentials of the functions attaining such maximum at that point (Boyd and Vandenberghe, 2007). With this, we obtain that

$$\left[\frac{\partial||W||_\infty}{\partial W}\right]_k = \left\{\begin{array}{lll} 0 & \text{if} & |W_k| < \max_j\{|W_j|\}, \\ \tau_i & \text{if} & W_k = \max_j\{|W_j|\}, \\ -\tau_i & \text{if} & -W_k = \max_j\{|W_j|\}, \end{array}\right.$$

with τ_i the convex hull coefficients, i.e.,

$$\sum_{i\in I}\tau_i = 1, \;\; I \equiv \left\{i : |W_i| = \max_j\{|W_j|\}\right\}.$$

Now, since the scale of $\frac{\partial||W||_\infty}{\partial W}$ is not relevant (only its orientation) and by picking only the most convenient subgradient we arrive at

$$\left[\frac{\partial||W||_\infty}{\partial W}\right]_k = \left\{\begin{array}{ll} 0 & \text{if} \quad |W_k| < \max_j\{|W_j|\}, \\ \text{sign}(W_i) & \text{if} \quad |W_k| = \max_j\{|W_j|\}. \end{array}\right.$$

The same comments than those for norm ℓ_1 apply here; if needed, the ℓ_∞ norm can be approximated by a large norm such as ℓ_{100}.

Appendix D. General $\ell_{p\geq 1}$ RCH-NPP Solver

The generalized $\ell_{p\geq 1}$ RCH-NPP problem takes the form

$$\min_{X_+\in\mathcal{U}_+, X_-\in\mathcal{U}_-} \|X_+ - X_-\|_p, \tag{31}$$

for $p \geq 1$ and sets $\mathcal{U}_\pm$ defined as in Proposition 2. Such problem is an instance of a common family of problems arising in machine learning in the form

$$\min_x f(x) + r(x),$$

for f convex and differentiable, r convex and lower semicontinuous, but not necessarily differentiable. Such problems are addressed efficiently by making use of a proximal method (see Combettes and Pesquet 2009 for a thorough review), as long as two basic ingredients are provided: a subroutine to compute the gradient of f and an efficient method to solve the proximity operator of r, an optimization subproblem taking the form

$$\text{prox}_r(y) \;\equiv\; \min_x \frac{1}{2}||x-y||_2^2 + r(x).$$

Problem 31 can be written in $\min_x f(x) + r(x)$ form by defining

$$\begin{array}{rcl} x & = & \left[\begin{array}{c} X_+ \\ X_- \end{array}\right], \\ f(x) & = & \|X_+ - X_-\|_p, \\ r(x) & = & \iota_{\mathcal{U}_+}(X_+) + \iota_{\mathcal{U}_-}(X_-), \end{array}$$

where $\iota_{\mathcal{C}}(x)$ is an indicator function valued 0 if $x \in \mathcal{C}$, $+\infty$ else. Using the results of the previous appendix the gradient of f can be shown to take the form

$$\nabla f(x) = \begin{bmatrix} \left(\frac{|X_+ - X_-|}{||X_+ - X_-||_q}\right)^{q-1} \operatorname{sign}(X_+ - X_-) \\ -\left(\frac{|X_+ - X_-|}{||X_+ - X_-||_q}\right)^{q-1} \operatorname{sign}(X_+ - X_-) \end{bmatrix},$$

while the proximity operator of r is

$$\begin{aligned} \operatorname{prox}_r(y) &\equiv \min_x \frac{1}{2}||x-y||_2^2 + \iota_{\mathcal{U}_+}(X_+) + \iota_{\mathcal{U}_-}(X_-), \\ &= \left\{\min_{X_+} \frac{1}{2}||X_+ - Y_+||_2^2 + \iota_{\mathcal{U}_+}(X_+)\right\} + \left\{\min_{X_-} \frac{1}{2}||X_- - Y_-||_2^2 + \iota_{\mathcal{U}_-}(X_-)\right\}, \\ &= \left\{\min_{X_+ \in \mathcal{U}_+} \frac{1}{2}||X_+ - Y_+||_2^2\right\} + \left\{\min_{X_- \in \mathcal{U}_-} \frac{1}{2}||X_- - Y_-||_2^2\right\}, \end{aligned}$$

where y has also been decomposed in two parts Y_+ and Y_-. It is evident now that the proximity operator can be computed by solving two independent subproblems, which turn out to be instances of the classic RCH–NPP where one of the hulls is a singleton $Y_\pm$. Such problem is solved through trivial modifications of a standard RCH–NPP solver.

In our RapMinos implementation we make use of the FISTA proximal algorithm (Beck and Teboulle, 2009), which by the inclusion of the aforementioned gradient and proximity subroutines results in an effective $\ell_{p\geq 1}$ RCH-NPP solver.

It is also worth pointing out that for the extreme ℓ_1 and ℓ_∞ cases problem (31) becomes non–differentiable, preventing the use of the presented approach. Still, a solution is easily attainable by realizing that in these two cases the minimization of the norm function can be rewritten as a set of linear constraints, as

$$\begin{aligned} \min_x ||x||_1 &= \min_x \sum_i \max\{x_i, -x_i\} = \min_{x,z} \sum_i z_i \quad \text{s.t.} \quad z_i \geq x_i, -x_i \; \forall\, i, \\ \min_x ||x||_\infty &= \min_x \max\{|x_1|, \ldots, |x_d|\} = \min_{x,z} z \quad \text{s.t.} \quad z \geq x_i, -x_i \; \forall\, i. \end{aligned}$$

Hence, the whole problem is rewritten as a Linear Program, which we solve by making use of Matlab's internal LP solver routine *linprog*.

Appendix E. Bias Computation in ERCH–NPP

When no reduction of the hulls is applied in RCH–NPP the usual procedure to compute the bias is to take it in such a way that the classification hyperplane lies at the middle of the extreme points in the convex–hulls ($b = -\frac{1}{2}W \cdot (X_+ + X_-)$ for the optimal solution X_+ and X_- of Eq. 7). However, such bias value is not necessarily equivalent to the one obtained when solving ν–SVM, as already pointed out by Crisp and Burges (2000). The same situation holds for Eν–SVM, and so we show here how to compute the correct value of b.

The KKT complementary slackness conditions of the inner minimization problem in ERCH–Margin (Eq. 5) are the following

$$\begin{aligned} \lambda_i(W \cdot X_i - \alpha + \xi_i) = 0 \qquad & \forall\, i \in M_+, \\ \lambda_i(W \cdot X_i - \beta - \xi_i) = 0 \qquad & \forall\, i \in M_-, \\ \xi_i \mu_i = 0 \qquad & \forall\, i, \end{aligned}$$

from which, together with the relationships obtained from the derivatives of the Lagrangian (Eq. 8) the following statements can be derived

- If $i \in M_+, \lambda_i > 0 \implies W \cdot X_i - \alpha + \xi_i = 0$.
- If $i \in M_-, \lambda_i > 0 \implies W \cdot X_i - \beta - \xi_i = 0$.
- If $\lambda_i < \eta \implies \mu_i > 0 \implies \xi_i = 0$.

Joining these three facts we can compute α by finding an $i \in M_+$ s.t. $0 < \lambda_i < \eta$, as for this case $W \cdot X_i - \alpha = 0$, and similarly for β, obtaining

$$\begin{aligned} \alpha = W \cdot X_i \qquad & \text{for some } i \in M_+, 0 < \lambda_i < \eta, \\ \beta = W \cdot X_i \qquad & \text{for some } i \in M_-, 0 < \lambda_i < \eta. \end{aligned}$$

Once α and β are known the bias can be computed through the definitions of these two terms (see the proof for Proposition 1), as

$$b = -\frac{1}{2}(\alpha + \beta). \tag{32}$$

Therefore, for any given W in ERCH–Margin or ERCH–NPP its corresponding bias can be computed with the obtained formula. A similar derivation was already proposed in Chang and Lin (2001) for the ν–SVM, though the connection with RCH–Margin was not made.

It should be noted, however, that the presented bias computation requires the sets $i \in M_+, 0 < \lambda_i < \eta$ and $i \in M_-, 0 < \lambda_i < \eta$ to be non–empty. If one of them turns out to be empty, which is a not so uncommon situation in practice, the bias cannot be computed in closed form. In such cases lower and upper bounds on b can be derived from the KKT conditions, as done in Chang and Lin (2001). We follow such procedure to obtain bounds on b and pick some value in the admissible range. Another possible solution would be to determine the bias as the one maximizing classification accuracy over the training set, that is

$$b^* = \arg\max_b \sum_{i \in M} \text{sign}\left\{y_i(X_i \cdot W + b)\right\}.$$

Such problem is solvable in log–linear time by sorting all the $X_i \cdot W$ values and counting the number of correct labellings for each possible b between all couples of consecutive $X_i \cdot W$ values. Even though this procedure seems to be more solid than selecting b from some loose bounds, it is actually prone to overfitting. Only in settings where the training data presents low noise have we found this procedure to produce better test accuracies, and thus we recommend resorting instead to the bounds provided by the KKT conditions.

References

A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal of Imaging Sciences*, 2(1):183–202, 2009.

K.P. Bennett and E.J. Bredensteiner. Duality and geometry in SVM classifiers. In *Proceedings of the 17th International Conference on Machine Learning*, pages 57–64, 2000.

D.P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1995.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

S. Boyd and L. Vandenberghe. Subgradients. Notes for EE364b, Stanford University, Winter 2006-07, January 2007.

C.-C. Chang and C.-J. Lin. *LIBSVM: a Library for Support Vector Machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

F. H. Clarke. *Optimization and Nonsmooth Analysis*. Classics in Applied Mathematics. SIAM, 1990.

P.L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. *arXiv:0912.3522*, 2009.

C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

D.J. Crisp and C.J.C. Burges. A geometric interpretation of ν–SVM classifiers. In *Advances in Neural Information Processing Systems*, volume 12, 2000.

R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. Wiley-Interscience. Wiley & Sons, New York, 2nd edition, 2001.

C. Ericson. The Gilbert-Johnson-Keerthi algorithm. Technical report, Sony Computer Entertainment America, 2005.

D.G. De Figueiredo and L.A. Karlovitz. On the radial projection in normed spaces. *Bulletin of the American Mathematical Society*, 1967.

K. Huang, D. Zheng, I. King, and M.R. Lyu. Arbitrary norm support vector machines. *Neural Computation*, 21(2):560–582, 2009.

S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy. A fast iterative nearest point algorithm for support vector machine classifier design. *IEEE Transactions on Neural Networks*, 11(1):124–136, 2000.

J. López, Á. Barbero, and J.R. Dorronsoro. On the equivalence of the SMO and MDM algorithms for SVM training. In *Lecture Notes in Computer Science: Machine Learning and Knowledge Discovery in Databases*, volume 5211, pages 288–300. Springer, 2008.

J. López, Á. Barbero, and J.R. Dorronsoro. Clipping algorithms for solving the nearest point problem over reduced convex hulls. *Pattern Recognition*, 44(3):607–614, 2011a.

J. López, K. De Brabanter, J.R. Dorronsoro, and JAK Suykens. Sparse LS-SVMs with ℓ_0–norm minimization. ESANN, 2011b.

D.G. Luenberger and Y. Ye. *Linear and Nonlinear Programming*. Springer, 2008.

M.E. Mavroforakis and S. Theodoridis. A geometric approach to support vector machine (SVM) classification. *IEEE Transactions on Neural Networks*, 17(3):671–682, 2006.

M.E. Mavroforakis, M. Sdralis, and S. Theodoridis. A geometric nearest point algorithm for the efficient solution of the SVM classification task. *IEEE Transactions on Neural Networks*, 18(5):1545–1549, 2007.

F. Pérez-Cruz, J. Weston, D.J.L. Hermann, and B. Schölkopf. Extension of the ν–SVM range for classification. In *Advances in Learning Theory: Methods, Models and Applications*, volume 190, pages 179–196, 2003.

G. Rätsch. *Benchmark Repository*, 2000. Datasets available at `http://www.raetschlab.org/Members/raetsch/benchmark`.

R.T. Rockafellar. *Convex Analysis*, volume 28 of *Princeton Mathematics Series*. Princeton University Press, 1970.

R.T. Rockafellar and S. Uryasev. Conditional value-at-risk for general loss distributions. *Journal of Banking & Finance*, 26(7):1443–1472, 2002.

B. Schölkopf, A.J. Smola, R.C. Williamson, and P.L. Bartlett. New support vector algorithms. *Neural Computation*, 12(5):1207–1245, 2000.

Y. Shi, Y. Tian, G. Kou, Y. Peng, and J. Li. Feature selection via ℓ_p–norm support vector machines. In *Optimization Based Data Mining: Theory and Applications*, pages 107–116. Springer, 2011.

A. Takeda and M. Sugiyama. ν–support vector machine as conditional value-at-risk minimization. In *Proceedings of the 25th International Conference on Machine Learning*, pages 1056–1063, 2008.

A. Takeda and M. Sugiyama. On generalization and non–convex optimization of extended ν–support vector machine. *New Generation Computing*, 27:259–279, 2009.

A. Takeda, H. Mitsugi, and T. Kanamori. A unified classification model based on robust optimization. *Neural Computation*, 25 (3):759–804, 2013.

J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani. 1–norm support vector machines. In S. Thrun, L.K. Saul, and B. Schölkopf, editors, *Neural Information Processing Systems*. MIT Press, 2003.

Journal of Machine Learning Research 16 (2015) 371-416 Submitted 8/13; Revised 4/14; Published 3/15

Composite Self-Concordant Minimization

Quoc Tran-Dinh QUOC.TRANDINH@EPFL.CH
Anastasios Kyrillidis ANASTASIOS.KYRILLIDIS@EPFL.CH
Volkan Cevher VOLKAN.CEVHER@EPFL.CH
Laboratory for Information and Inference Systems (LIONS)
École Polytechnique Fédérale de Lausanne (EPFL)
CH1015-Lausanne, Switzerland

Editor: Benjamin Recht

Abstract

We propose a variable metric framework for minimizing the sum of a self-concordant function and a possibly non-smooth convex function, endowed with an easily computable proximal operator. We theoretically establish the convergence of our framework without relying on the usual Lipschitz gradient assumption on the smooth part. An important highlight of our work is a new set of analytic step-size selection and correction procedures based on the structure of the problem. We describe concrete algorithmic instances of our framework for several interesting applications and demonstrate them numerically on both synthetic and real data.

Keywords: proximal-gradient/Newton method, composite minimization, self-concordance, sparse convex optimization, graph learning

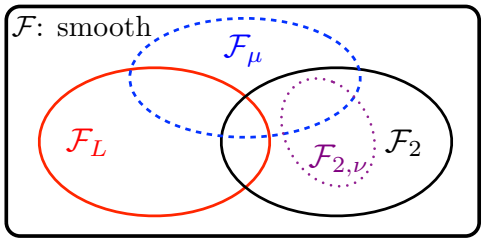
1. Introduction

The literature on the formulation, analysis, and applications of *composite convex minimization* is ever expanding due to its broad applications in machine learning, signal processing, and statistics. By composite minimization, we refer to the following optimization problem:

$$F^* := \min_{\mathbf{x}\in\mathbb{R}^n} \left\{ F(\mathbf{x}) \;\; | \;\; F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x}) \right\}, \tag{1}$$

where f and g are both closed and convex, and n is the problem dimension. In the canonical setting of the composite minimization problem (1), the functions f and g are assumed to be smooth and non-smooth, respectively (Nesterov, 2007). Such composite objectives naturally arise, for instance, in maximum a posteriori model estimation, where we regularize a model likelihood function as measured by a data-driven smooth term f with a non-smooth model prior g, which carries some notion of model complexity (e.g., sparsity, low-rankness, etc.).

In theory, many convex problem instances of the form (1) have a well-understood structure, and hence high accuracy solutions can be efficiently obtained with polynomial time methods, such as interior point methods (IPM) after transforming them into conic quadratic programming or semidefinite programming formulations (Ben-Tal and Nemirovski, 2001; Grant et al., 2006; Nesterov and Nemirovski, 1994). *In practice*, however, the curse-of-dimensionality renders these methods impractical for large-scale problems. Moreover, the



Class	**Property** $\mathbf{x},\mathbf{y}\in\mathrm{dom}(f),\ \mathbf{v}\in\mathbb{R}^n, 0\leq\mu\leq L<+\infty$
$\mathcal{F}_L$	$\|\nabla f(\mathbf{x})-\nabla f(\mathbf{y})\|_*\leq L\|\mathbf{x}-\mathbf{y}\|$
$\mathcal{F}_\mu$	$\frac{\mu}{2}\|\mathbf{x}-\mathbf{y}\|^2+f(\mathbf{x})+\nabla f(\mathbf{x})^T(\mathbf{y}-\mathbf{x})\leq f(\mathbf{y})$
$\mathcal{F}_2$	$\|\varphi'''(t)\|\leq 2\varphi''(t)^{3/2}$: $\varphi(t)=f(\mathbf{x}+t\mathbf{v}),\ t\in\mathbb{R}$
$\mathcal{F}_{2,\nu}$	$\mathcal{F}_2$ and $\sup_{\mathbf{v}\in\mathbb{R}^n}\left\{2\nabla f(\mathbf{x})^T\mathbf{v}-\|\mathbf{v}\|^2_{\mathbf{x}}\right\}\leq\nu$

Figure 1: Common structural assumptions on the smooth function f.

presence of a non-smooth term g prevents direct applications of scalable smooth optimization techniques, such as sequential linear or quadratic programming.

Fortunately, we can provably trade-off accuracy with computation by further exploiting the individual structures of f and g. Existing methods invariably rely on two structural assumptions that particularly stand out among many others. First, we often assume that f has Lipschitz continuous gradient (i.e., $f\in\mathcal{F}_L$: cf., Figure 1). Second, we assume that the proximal operator of g ($\mathrm{prox}_g^{\mathbf{H}}(\mathbf{y}):=\arg\min_{\mathbf{x}\in\mathbb{R}^n}\left\{g(\mathbf{x})+(1/2)\|\mathbf{x}-\mathbf{y}\|^2_{\mathbf{H}}\right\}$) is, in a user-defined sense, easy to compute for some $\mathbf{H}\succ 0$ (e.g., $\mathbf{H}$ is diagonal); i.e., we can computationally afford to apply the proximal operator in an iterative fashion. In this case, g is said to be "*tractably proximal*". On the basis of these structures, we can design algorithms featuring a full spectrum of (nearly) dimension-independent, global convergence rates with well-understood analytical complexity (see Table 1).

Order	Method example	Main oracle	Analytical complexity
1-st	[Accelerated]^a-[proximal]-gradient^b	$\nabla f, \mathrm{prox}_g^{L\mathbb{I}_n}$	$[\mathcal{O}(\epsilon^{-1/2})]$ $\mathcal{O}(\epsilon^{-1})$
1^+-th	Proximal-quasi-Newton^c	$\mathbf{H}_k, \nabla f, \mathrm{prox}_g^{\mathbf{H}_k}$	$\mathcal{O}(\log\epsilon^{-1})$ or faster
2-nd	Proximal-Newton^d	$\nabla^2 f, \nabla f, \mathrm{prox}_g^{\nabla^2 f}$	$\mathcal{O}(\log\log\epsilon^{-1})$[local]

See (Beck and Teboulle, 2009a)^{a,b},(Becker and Fadili, 2012)^c,(Lee et al., 2012)^d,(Nesterov, 2004, 2007)^{a,b}.

Table 1: Taxonomy of [accelerated] [proximal]-gradient methods when $f\in\mathcal{F}_L$ or proximal-[quasi]-Newton methods when $f\in\mathcal{F}_L\cap\mathcal{F}_\mu$ to reach an ε-solution (e.g., $F(\mathbf{x}^k)-F^*\leq\epsilon$).

Unfortunately, existing algorithms have become inseparable with the Lipschitz gradient assumption on f and are still being applied to solve (1) in applications where this assumption does not hold. For instance, when $\mathrm{prox}_g^{\mathbf{H}}(\mathbf{y})$ is not easy to compute, it is still possible to establish convergence—albeit slower—with smoothing, splitting or primal-dual decomposition techniques (Chambolle and Pock, 2011; Eckstein and Bertsekas, 1992; Nesterov, 2005a,b; Tran-Dinh et al., 2013c). However, when $f\notin\mathcal{F}_L$, the composite problems of the form (1) are not within the full theoretical grasp. In particular, there is no known global convergence rate. One kludge to handle $f\notin\mathcal{F}_L$ is to use sequential quadratic approximation of f to reduce the subproblems to the Lipschitz gradient case. For local convergence of these methods, we need *strong regularity* assumptions on f (i.e., $\mu\mathbb{I}\preceq\nabla^2 f(\mathbf{x})\preceq L\mathbb{I}$) near

the optimal solution. Attempts at global convergence require a *globalization strategy* such as line search procedures (cf., Section 1.2). However, neither the strong regularity nor the line search assumptions can be certified *a priori*.

To this end, we address the following question in this paper: "Is it possible to efficiently solve non-trivial instances of (1) for non-global Lipschitz continuous gradient f with rigorous global convergence guarantees?" The answer is positive (at least for a broad class of functions): We can still cover a full spectrum of global convergence rates with well-characterizable computation and accuracy trade-offs (akin to Table 1 for $f \in \mathcal{F}_L$) for self-concordant f (in particular, self-concordant barriers) (Nemirovskii and Todd, 2008; Nesterov and Nemirovski, 1994):

Definition 1 (Self-concordant (barrier) functions) *A convex function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be self-concordant (i.e., $f \in \mathcal{F}_M$) with parameter $M \geq 0$, if $|\varphi'''(t)| \leq M\varphi''(t)^{3/2}$, where $\varphi(t) := f(\mathbf{x} + t\mathbf{v})$ for all $t \in \mathbb{R}$, $\mathbf{x} \in \mathrm{dom}(f)$ and $\mathbf{v} \in \mathbb{R}^n$ such that $\mathbf{x} + t\mathbf{v} \in \mathrm{dom}(f)$. When $M = 2$, the function f is said to be a standard self-concordant, i.e., $f \in \mathcal{F}_2$.[1] A standard self-concordant function $f \in \mathcal{F}_2$ is a ν-self-concordant barrier of a given convex set Ω with parameter $\nu > 0$, i.e., $f \in \mathcal{F}_{2,\nu}$, when φ also satisfies $|\varphi'(t)| \leq \sqrt{\nu}\varphi''(t)^{1/2}$ and $f(\mathbf{x}) \to +\infty$ as $\mathbf{x} \to \partial\Omega$, the boundary of Ω.*

While there are other definitions of self-concordant functions and self-concordant barriers (Boyd and Vandenberghe, 2004; Nemirovskii and Todd, 2008; Nesterov and Nemirovski, 1994; Nesterov, 2004), we use Definition 1 in the sequel, unless otherwise stated.

1.1 Why is the Assumption $f \in \mathcal{F}_2$ Interesting for Composite Minimization?

The assumption $f \in \mathcal{F}_2$ in (1) is quite natural for two reasons. First, several important applications directly feature a self-concordant f, which does not have global Lipschitz continuous gradient. Second, self-concordant composite problems can enable approximate solutions of general constrained convex problems where the constraint set is endowed with a ν-self-concordant barrier function.[2] Both settings clearly benefit from scalable algorithms. Hence, we now highlight three examples below, based on compositions with the log-functions. Keep in mind that this list of examples is not meant to be exhaustive.

Log-determinant: The matrix variable function $f(\mathbf{\Theta}) := -\log\det\mathbf{\Theta}$ is self-concordant with $\mathrm{dom}(f) := \{\mathbf{\Theta} \in \mathbb{S}^p \mid \mathbf{\Theta} \succ 0\}$, where $\mathbb{S}^p$ is the set of $p \times p$ symmetric matrices. As a stylized application, consider learning a Gaussian Markov random field (GMRF) of p nodes/variables from a data set $\mathcal{D} := \{\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_m\}$, where $\boldsymbol{\phi}_j \in \mathcal{D}$ is a p-dimensional random vector with Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$. Let $\mathbf{\Theta} := \mathbf{\Sigma}^{-1}$ be the inverse covariance (or the precision) matrix for the model. To satisfy the conditional dependencies with respect to the GMRF, $\mathbf{\Theta}$ must have zero in $(\mathbf{\Theta})_{ij}$ corresponding to the absence of an edge between node i and node j; cf., (Dempster, 1972).

1. We use this constant for convenience in the derivations since if $f \in \mathcal{F}_M$, then $(M^2/4)f \in \mathcal{F}_2$.
2. Let us consider a constrained convex minimization $\mathbf{x}^*_C := \arg\min_{\mathbf{x}\in C} g(\mathbf{x})$, where the feasible convex set C is endowed with a ν-self-concordant barrier $\Psi_C(\mathbf{x})$. If we let $f(\mathbf{x}) := \frac{\epsilon}{\nu}\Psi_C(\mathbf{x})$, then the solution $\mathbf{x}^*$ of the composite minimization problem (1) well-approximates $\mathbf{x}^*_C$ as $g(\mathbf{x}^*) \leq g(\mathbf{x}^*_C) + (\nabla f(\mathbf{x}^*) + \partial g(\mathbf{x}^*))^T(\mathbf{x}^* - \mathbf{x}^*_C) + \epsilon$. The middle term can be controlled by accuracy at which we solve the composite minimization problem (Nesterov, 2007, 2011).

We can learn GMRFs with theoretical guarantees from as few as $\mathcal{O}(d^2 \log p)$ data samples, where d is the graph node degree, via ℓ_1-norm regularization formulation (see Ravikumar et al. 2011):

$$\boldsymbol{\Theta}^* := \arg\min_{\boldsymbol{\Theta} \succ 0} \Big\{ \underbrace{-\log\det(\boldsymbol{\Theta}) + \mathrm{tr}(\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Theta})}_{=:f(\boldsymbol{\Theta})} + \underbrace{\rho \|\mathbf{vec}(\boldsymbol{\Theta})\|_1}_{=:g(\boldsymbol{\Theta})} \Big\}, \tag{2}$$

where $\rho > 0$ parameter balances a Gaussian model likelihood and the sparsity of the solution, $\widehat{\boldsymbol{\Sigma}}$ is the empirical covariance estimate, and **vec** is the vectorization operator. The formulation also applies for learning models beyond GMRFs, such as the Ising model, since $f(\boldsymbol{\Theta})$ acts also as a Bregman distance (Banerjee et al., 2008).

Numerical solution methods for solving problem (2) have been extensively studied, e.g. in (Banerjee et al., 2008; Hsieh et al., 2011; Lee et al., 2012; Lu, 2010; Olsen et al., 2012; Rolfs et al., 2012; Scheinberg and Rish, 2009; Scheinberg et al., 2010; Yuan, 2012). However, none so far exploits $f \in \mathcal{F}_{2,\nu}$ and feature global convergence guarantees: cf., Sect. 1.2.

Log-barrier for linear inequalities: The function $f(\mathbf{x}) := -\log(\mathbf{a}^T\mathbf{x} - b)$ is a self-concordant barrier with $\mathrm{dom}(f) := \big\{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{a}^T\mathbf{x} > b\big\}$. As a stylized application, consider the low-light imaging problem in signal processing (Harmany et al., 2012), where the imaging data is collected by counting photons hitting a detector over the time. In this setting, we wish to accurately reconstruct an image in low-light, which leads to noisy measurements due to low photon count levels. We can express our observation model using the Poisson distribution as

$$\mathbb{P}(\mathbf{y}|\mathcal{A}(\mathbf{x})) = \prod_{i=1}^{m} \frac{(\mathbf{a}_i^T\mathbf{x})^{y_i}}{y_i!} e^{-\mathbf{a}_i^T\mathbf{x}},$$

where $\mathbf{x}$ is the true image, $\mathcal{A}$ is a linear operator that projects the scene onto the set of observations, $\mathbf{a}_i$ is the i-th row of $\mathcal{A}$, and $\mathbf{y} \in \mathbb{Z}_+^m$ is a vector of observed photon counts.

Via the log-likelihood formulation, we stumble upon a composite minimization problem:

$$\mathbf{x}^* := \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Big\{ \underbrace{\sum_{i=1}^{m} \mathbf{a}_i^T\mathbf{x} - \sum_{i=1}^{m} y_i \log(\mathbf{a}_i^T\mathbf{x})}_{=:f(\mathbf{x})} + g(\mathbf{x}) \Big\}, \tag{3}$$

where $f(\mathbf{x})$ is self-concordant (but not standard). In the above formulation, the typical image priors $g(\mathbf{x})$ include the ℓ_1-norm for sparsity in a known basis, total variation seminorm of the image, and the positivity of the image pixels. While the formulation (3) seems specific to imaging, it is also common in sparse regression with unknown noise variance (Städler et al., 2012), heteroscedastic LASSO (Dalalyan et al., 2013), barrier approximations of, e.g., the Dantzig selector (Candes and Tao, 2007) and quantum tomography (Banaszek et al., 1999) as well.

The current state of the art solver is called SPIRAL-TAP (Harmany et al., 2012), which biases the logarithmic term (i.e., $\log(\mathbf{a}_i^T\mathbf{x} + \varepsilon) \to \log(\mathbf{a}_i^T\mathbf{x})$, where $\varepsilon \ll 1$) and then applies non-monotone composite gradient descent algorithms for $\mathcal{F}_L$ with a Barzilai-Borwein stepsize as well as other line-search strategies.

Logarithm of concave quadratic functions: The function $f(\mathbf{x}) := -\log\big(\sigma^2 - \|\mathbf{A}\mathbf{x}-\mathbf{y}\|_2^2\big)$ is self-concordant with $\mathrm{dom}(f) := \big\{\mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 < \sigma^2\big\}$. As a stylized application,

we consider the basis pursuit denoising (BPDN) formulation (van den Berg and Friedlander, 2008) as

$$\mathbf{x}^* := \arg\min_{\mathbf{x}\in\mathbb{R}^n} \left\{ g(\mathbf{x}) \mid \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 \leq \sigma^2 \right\}. \tag{4}$$

The BPDN criteria is commonly used in magnetic resonance imaging (MRI) where $\mathbf{A}$ is a subsampled Fourier operator, $\mathbf{y}$ is the MRI scan data, and σ^2 is a known machine noise level (i.e., obtained during a pre-scan). In (4), g is an image prior, e.g., similar to the Poisson imaging problem. Approximate solutions to (4) can be obtained via a barrier formulation

$$\mathbf{x}_t^* := \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Big\{ \underbrace{-t\log\left(\sigma^2 - \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2\right)}_{=:f(\mathbf{x})} \; + \; g(\mathbf{x}) \Big\}, \tag{5}$$

where $t > 0$ is a penalty parameter which controls the quality of the approximation. The BPDN formulation is quite generic and has several other applications in statistical regression, geophysics, and signal processing.

Several different approaches solve the BPDN problem (4), some of which require projections onto the constraint set, including Douglas-Rachford splitting, proximal methods, and the SPGL$_1$ method (van den Berg and Friedlander, 2008; Combettes and Wajs, 2005).

1.2 Related Work

Our attempt is to briefly describe the work that revolves around (1) with the main assumptions of $f \in \mathcal{F}_L$ and the proximal operator of g being computationally tractable. In fact, Douglas-Rachford splitting methods can obtain numerical solutions to (1) when the self-concordant functions are endowed with tractable proximal maps. However, it is computationally easier to calculate the gradient of $f \in \mathcal{F}_2$ than their proximal maps.

One of the main approaches in this setting is based on operator splitting. By presenting the optimality condition of problem (1) as an inclusion of two monotone operators, one can apply splitting techniques, such as forward-backward or Douglas-Rachford methods, to solve the resulting monotone inclusion (Briceno-Arias and Combettes, 2011; Facchinei and Pang, 2003; Goldstein and Osher, 2009). In our context, several variants of this approach have been studied. For example, projected gradient or proximal-gradient methods and fast proximal-gradient methods have been considered, see, e.g., (Beck and Teboulle, 2009a; Mine and Fukushima, 1981; Nesterov, 2007). In all these methods, the main assumption required to prove the convergence is the global Lipschitz continuity of the gradient of the smooth function f. Unfortunately, when $f \notin \mathcal{F}_L$ but $f \in \mathcal{F}_2$, these theoretical results on the global convergence and the global convergence rates are no longer applicable.

Other mainstream approaches for (1) include augmented Lagrangian and alternating techniques: cf., (Boyd et al., 2011; Goldfarb and Ma, 2012). These methods have empirically proven to be quite powerful in specific applications. The main disadvantage of these methods is the manual tuning of the penalty parameter in the augmented Lagrangian function, which is not yet well-understood for general problems. Consequently, the analysis of global convergence as well as the convergence rate is an issue since the performance of the algorithms strongly depends on the choice of this penalty parameter in practice. Moreover, as indicated in a recent work (Goldstein et al., 2012), alternating direction methods of multipliers as well as alternating linearization methods can be viewed as splitting methods in

the convex optimization context. Hence, it is unclear if this line of work is likely to lead to any rigorous guarantees when $f \in \mathcal{F}_2$.

An emerging direction for solving composite minimization problems (1) is based on the proximal-Newton method. The origins of this method can be traced back to the work of (Bonnans, 1994), which relies on the concept of *strong regularity* introduced by (Robinson, 1980) for generalized equations. In the convex case, this method has been studied by several authors such as (Becker and Fadili, 2012; Lee et al., 2012; Schmidt et al., 2011). So far, methods along this line are applied to solve a generic problem of the form (1) even when $f \in \mathcal{F}_2$. The convergence analysis of these methods is encouraged by standard Newton methods and requires the strong regularity of the Hessian of f near the optimal solution (i.e., $\mu\mathbb{I} \preceq \nabla^2 f(\mathbf{x}) \preceq L\mathbb{I}$). This assumption used in (Lee et al., 2012) is stronger than assuming $\nabla^2 f(\mathbf{x}^*)$ to be positive definite at the solution $\mathbf{x}^*$ as in our approach below. Moreover, the global convergence can only be proved by applying a certain globalization strategy such as line-search (Lee et al., 2012) or trust-region. Unfortunately, none of these assumptions can be verified before the algorithm execution for the intended applications. By exploiting the self-concordance concept, we can show the global convergence of proximal-Newton methods without any globalization strategy (e.g., line search or trust-region approach).

1.3 Our Contributions

Interior point methods are always an option while solving the self-concordant composite problems (1) numerically by means of disciplined convex programming (Grant et al., 2006; Löfberg, 2004). More concretely, in the IPM setting, we set up an equivalent problem to (1) that typically avoids the non-smooth term $g(x)$ in the objective by lifting the problem dimensions with slack variables and introducing additional constraints. The new constraints may then be embedded into the objective through a barrier function. We then solve a sequence of smooth problems (e.g., with Newton methods) and "path-follow"[3] to obtain an accurate solution (Nemirovskii and Todd, 2008; Nesterov, 2004). In this loop, many of the underlying structures within the original problem, such as sparsity, can be lost due to pre-conditioning or Newton direction scaling (e.g., Nesterov-Todd scaling, Nesterov and Todd 1997). The efficiency and the memory bottlenecks of the overall scheme then heavily depends on the workhorse algorithm that solves the smooth problems.

In stark contrast, we introduce an algorithmic framework that directly handles the composite minimization problem (1) without increasing the original problem dimensions. For problems of larger dimensions, this is the main argument in favor of our approach. Instead of solving a sequence of smooth problems, we solve a sequence of non-smooth proximal problems with a variable metric (i.e., our workhorse). Fortunately, these proximal problems feature the composite form (1) with a Lipschitz gradient (and oft-times strongly convex) smooth term. Hence, we leverage the tremendous amount of research (cf., Table 1) done over the last decades. Surprisingly, we can even retain the original problem structures that lead to computational ease in many cases (e.g., see Section 4.1).

Our specific contributions can be summarized as follows:

1. We develop a *variable metric* framework for minimizing the sum $f + g$ of a self-concordant function f and a convex, possibly nonsmooth function g. Our approach

3. It is also referred to as a homotopy method.

relies on the solution of a convex subproblem obtained by linearizing and regularizing the first term f. To achieve monotonic descent, we develop a new set of *analytic* step-size selection and correction procedures based on the structure of the problem.

2. We establish both the global and the local convergence of different variable metric strategies. We first derive an expected result: when the variable metric is the Hessian $\nabla^2 f(\mathbf{x}^k)$ of f at iteration k, the resulting algorithm locally exhibits quadratic convergence rate within an explicit region. We then show that variable metrics satisfying the Dennis-Moré-type condition (Dennis and Moré, 1974) exhibit superlinear convergence.

3. We pay particular attention to diagonal variable metrics as many of the proximal subproblems can be solved exactly (i.e., in closed form). We derive conditions on when these variants achieve locally linear convergence.

4. We apply our algorithms to the aforementioned real-world and synthetic problems to highlight the strengths and the weaknesses of our scheme. For instance, in the graph learning problem (2), our framework can avoid matrix inversions as well as Cholesky decompositions in learning graphs. In Poisson intensity reconstruction (3), up to around $80\times$ acceleration is possible over the state-of-the-art solver.

We highlight three key practical contributions to numerical optimization. First, in the proximal-Newton method, our analytical step-size procedures allow us to do away with any globalization strategy (e.g., line-search). This has a significant practical impact when the evaluation of the functions is expensive. We show how to combine the analytical step-size selection with the standard backtracking or forward line-search procedures to enhance the global convergence of our method. Our analytical quadratic convergence characterization helps us adaptively switch from *damped* step-size to a *full* step-size. Second, in the proximal-gradient method setting, we establish a step-size selection and correction mechanism. The step-size selection procedure can be considered as a predictor, where existing step-size rules that leverage local information can be used. The step-size corrector then adapts the local information of the function to achieve the best theoretical decrease in the objective function. While our procedure does not require any function evaluations, we can further enhance convergence whenever we are allowed function evaluations. Finally, our framework, as we demonstrate in (Tran-Dinh et al., 2014a), accommodates a path-following strategy, which enable us to approximately solve constrained non-smooth convex minimization problems with rigorous guarantees.

Paper outline. In Section 2, we first recall some fundamental concepts of convex optimization and self-concordant functions used in this paper. Section 3 presents our algorithmic framework using three different instances with convergence results, complexity estimates and modifications. Section 4 deals with three concrete instances of our algorithmic framework. Section 5 provides numerical experiments to illustrate the impact of the proposed methods. Section 6 concludes the paper.

2. Preliminaries

Notation: We reserve lower-case and bold lower-case letters for scalar and vector representation, respectively. Upper-case bold letters denote matrices. We denote $\mathbb{S}^p_+$ (reps., $\mathbb{S}^p_{++}$) for the set of symmetric positive definite (reps., positive semidefinite) matrices of size $p \times p$. For a proper, lower semicontinuous convex function f from $\mathbb{R}^n$ to $\mathbb{R} \cup \{+\infty\}$, we denote its domain by $\mathrm{dom}(f)$, i.e., $\mathrm{dom}(f) := \{\mathbf{x} \in \mathbb{R}^n \mid f(\mathbf{x}) < +\infty\}$ (see, e.g., Rockafellar 1970).

Weighted norm and local norm: Given a matrix $\mathbf{H} \in \mathbb{S}^n_{++}$, we define the weighted norm $\|\mathbf{x}\|_{\mathbf{H}} := \sqrt{\mathbf{x}^T\mathbf{H}\mathbf{x}}$, $\forall \mathbf{x} \in \mathbb{R}^n$; its dual norm is defined as $\|\mathbf{x}\|^*_{\mathbf{H}} := \max_{\|\mathbf{y}\|_{\mathbf{H}} \le 1} \mathbf{y}^T\mathbf{x} = \sqrt{\mathbf{x}^T\mathbf{H}^{-1}\mathbf{x}}$. If $\mathbf{H}$ is only positive semidefinite (i.e., $\mathbf{H} \in \mathbb{S}^n_+$), then $\|\mathbf{x}\|_{\mathbf{H}}$ reduces to a semi-norm. Let $f \in \mathcal{F}_2$ and $\mathbf{x} \in \mathrm{dom}(f)$ so that $\nabla^2 f(\mathbf{x})$ is positive definite. For a given vector $\mathbf{v} \in \mathbb{R}^n$, the local norm around $\mathbf{x} \in \mathrm{dom}(f)$ with respect to f is defined as $\|\mathbf{v}\|_{\mathbf{x}} := \left(\mathbf{v}^T\nabla^2 f(\mathbf{x})\mathbf{v}\right)^{1/2}$, while the corresponding dual norm is given by $\|\mathbf{v}\|^*_{\mathbf{x}} = \left(\mathbf{v}^T\nabla^2 f(\mathbf{x})^{-1}\mathbf{v}\right)^{1/2}$.

Subdifferential and subgradient: Given a proper, lower semicontinuous convex function, we define the subdifferential of g at $\mathbf{x} \in \mathrm{dom}(g)$ as

$$\partial g(\mathbf{x}) := \left\{\mathbf{v} \in \mathbb{R}^n \mid g(\mathbf{y}) - g(\mathbf{x}) \ge \mathbf{v}^T(\mathbf{y} - \mathbf{x}),\ \forall \mathbf{y} \in \mathrm{dom}(g)\right\}.$$

If $\partial g(\mathbf{x}) \neq \emptyset$ then each element in $\partial g(\mathbf{x})$ is called a subgradient of g at $\mathbf{x}$. In particular, if g is differentiable, we use $\nabla g(\mathbf{x})$ to denote its derivative at $\mathbf{x} \in \mathrm{dom}(g)$, and $\partial g(\mathbf{x}) \equiv \{\nabla f(\mathbf{x})\}$.

Proximity operator: A basic tool to handle the nonsmoothness of a convex function g is its proximity operator (or proximal operator) $\mathrm{prox}^{\mathbf{H}}_g$, whose definition is given in Section 1. For notational convenience in our derivations, we alter this definition in the sequel as follows: Let g be a proper lower semicontinuous and convex in $\mathbb{R}^n$ and $\mathbf{H} \in \mathbb{S}^n_+$. We define

$$P^g_{\mathbf{H}}(\mathbf{u}) := \arg\min_{\mathbf{x} \in \mathbb{R}^n} \left\{g(\mathbf{x}) + (1/2)\mathbf{x}^T\mathbf{H}\mathbf{x} - \mathbf{u}^T\mathbf{x}\right\}, \quad \forall \mathbf{u} \in \mathbb{R}^n, \tag{6}$$

as the proximity operator for the nonsmooth g, which has the following properties Hiriart-Urruty and Lemaréchal (2001).

Lemma 2 *Assume that $\mathbf{H} \in \mathbb{S}^n_{++}$. Then, the operator $P^g_{\mathbf{H}}$ in (6) is single-valued and satisfies the following property:*

$$(P^g_{\mathbf{H}}(\mathbf{u}) - P^g_{\mathbf{H}}(\mathbf{v}))^T(\mathbf{u} - \mathbf{v}) \ge \left\|P^g_{\mathbf{H}}(\mathbf{u}) - P^g_{\mathbf{H}}(\mathbf{v})\right\|^2_{\mathbf{H}}, \tag{7}$$

for all $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$. Consequently, $P^g_{\mathbf{H}}$ is a nonexpansive mapping, i.e.,

$$\left\|P^g_{\mathbf{H}}(\mathbf{u}) - P^g_{\mathbf{H}}(\mathbf{v})\right\|_{\mathbf{H}} \le \|\mathbf{u} - \mathbf{v}\|^*_{\mathbf{H}}. \tag{8}$$

Proof This lemma is already known in the literature, see, e.g., (Rockafellar, 1976). For the sake of completeness, we give a short proof here. The single-valuedness of $P^g_{\mathbf{H}}$ is obvious due to the strong convexity of the objective function in (6). Let $\boldsymbol{\xi}_{\mathbf{u}} := P^g_{\mathbf{H}}(\mathbf{u})$ and $\boldsymbol{\xi}_{\mathbf{v}} := P^g_{\mathbf{H}}(\mathbf{v})$. By the definition of $P^g_{\mathbf{H}}$, we have $\mathbf{u} - \mathbf{H}\boldsymbol{\xi}_{\mathbf{u}} \in \partial g(\boldsymbol{\xi}_{\mathbf{u}})$ and $\mathbf{v} - \mathbf{H}\boldsymbol{\xi}_{\mathbf{u}} \in \partial g(\boldsymbol{\xi}_{\mathbf{v}})$. Since g is convex, we have $(\mathbf{u} - \mathbf{H}\boldsymbol{\xi}_{\mathbf{u}} - (\mathbf{v} - \mathbf{H}\boldsymbol{\xi}_{\mathbf{v}}))^T(\boldsymbol{\xi}_{\mathbf{u}} - \boldsymbol{\xi}_{\mathbf{v}}) \ge 0$. This inequality leads to $(\mathbf{u} - \mathbf{v})^T(\boldsymbol{\xi}_{\mathbf{u}} - \boldsymbol{\xi}_{\mathbf{v}}) \ge (\boldsymbol{\xi}_{\mathbf{u}} - \boldsymbol{\xi}_{\mathbf{v}})^T\mathbf{H}(\boldsymbol{\xi}_{\mathbf{u}} - \boldsymbol{\xi}_{\mathbf{v}}) = \|\boldsymbol{\xi}_{\mathbf{u}} - \boldsymbol{\xi}_{\mathbf{v}}\|^2_{\mathbf{H}}$ which is indeed (7). Via the generalized Cauchy-Schwarz inequality, (7) leads to (8). ■

Key self-concordant bounds: Based on (Nesterov, 2004, Theorems 4.1.7 and 4.1.8), for a given standard self-concordant function f, we recall the following inequalities

$$\omega(\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}}) + \nabla f(\mathbf{x})^T(\mathbf{y}-\mathbf{x}) + f(\mathbf{x}) \leq f(\mathbf{y}), \tag{9}$$

$$f(\mathbf{y}) \leq f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y}-\mathbf{x}) + \omega_*(\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}}), \tag{10}$$

where $\omega : \mathbb{R} \to \mathbb{R}_+$ is defined as $\omega(t) := t - \ln(1+t)$ and $\omega_* : [0,1] \to \mathbb{R}_+$ is defined as $\omega_*(t) := -t - \ln(1-t)$. These functions are both nonnegative, strictly convex and increasing. Hence, (9) holds for all $\mathbf{x}, \mathbf{y} \in \mathrm{dom}(f)$, and (10) holds for all $\mathbf{x}, \mathbf{y} \in \mathrm{dom}(f)$ such that $\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}} < 1$. In contrast to the "global" inequalities for the function classes $\mathcal{F}_L$ and $\mathcal{F}_\mu$ (cf., Figure 1), the self-concordant inequalities are based on "local" quantities. Moreover, these bounds are no longer quadratic which prevents naive applications of the methods from $\mathcal{F}_{L,\mu}$.

Remark 3 *The proof of* (9)-(10) *is based on the condition* $\nabla^2 f(\mathbf{x}) \succ 0$ *for all* $\mathbf{x} \in \mathrm{dom}(f)$*, see (Nesterov, 2004). In this paper, we work with the function* f *defined by* $f(\mathbf{x}) := \varphi(\mathbf{A}\mathbf{x}+\mathbf{b})$*, where* φ *is a standard self-concordant function such that* $\nabla^2\varphi(\mathbf{u}) \succ 0$ *for all* $\mathbf{u} \in \mathrm{dom}(\varphi)$*. Therefore, we have* $\nabla^2 f(\mathbf{x}) = \mathbf{A}^T\nabla^2\varphi(\mathbf{A}\mathbf{x}+\mathbf{b})\mathbf{A}$*, which is possibly singular without further conditions on matrix* $\mathbf{A}$*. Consequently, the local norm* $\|\cdot\|_{\mathbf{x}}$ *defined via* $\nabla^2 f(\mathbf{x})$ *reduces to a* semi-norm*. However, the inequalities* (9)-(10) *still hold w.r.t. this semi-norm. Indeed, since* φ *is standard self-concordant with* $\nabla^2\varphi(\mathbf{u}) \succ 0$ *for all* $\mathbf{u} \in \mathrm{dom}(\varphi)$*, we have* $\varphi(\hat{\mathbf{u}}) \geq \varphi(\mathbf{u}) + \nabla\varphi(\mathbf{u})^T(\hat{\mathbf{u}}-\mathbf{u}) + \omega\big(\|\hat{\mathbf{u}}-\mathbf{u}\|_{\mathbf{u}}\big)$*. By substituting* $\mathbf{u} = \mathbf{A}\mathbf{x}+\mathbf{b} \in \mathrm{dom}(\varphi)$ *and* $\hat{\mathbf{u}} = \mathbf{A}\hat{\mathbf{x}}+\mathbf{b} \in \mathrm{dom}(\varphi)$*,* $(\mathbf{x}, \hat{\mathbf{x}} \in \mathrm{dom}(f))$ *into this inequality we obtain* $f(\hat{\mathbf{x}}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\hat{\mathbf{x}}-\mathbf{x}) + \omega(\|\hat{\mathbf{x}}-\mathbf{x}\|_{\mathbf{x}})$*, which is indeed* (9)*. The inequality* (10) *is proved similarly.*

3. Composite Self-Concordant Optimization

In this section, we propose a *variable metric* optimization framework that rigorously trades off computation and accuracy of solutions without transforming (1) into a higher dimension smooth convex optimization problem. We assume theoretically that the proximal subproblems can be solved exactly. However, our theory can be analyze for the inexact case, when we solve these problems up to a sufficiently high accuracy (typically, it is at least higher than (e.g., 0.1ε) the desired accuracy ε of (1) at the few last iterations), see, e.g., (Tran-Dinh et al., 2013b, 2014a). In our theoretical characterizations, we only rely on the following assumption:

Assumption A.1 *The function* f *is convex and standard self-concordant* (*see Definition 1*)*. The function* $g : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ *is proper, closed and convex.*

Under Assumption **A**.1, we have $\mathrm{dom}(F) = \mathrm{dom}(f) \cap \mathrm{dom}(g)$.

Unique solvability of (1) *and its optimality condition:* First, we show that problem (1) is uniquely solvable. The proof of this lemma can be done similarly as (Nesterov, 2004, Theorem 4.1.11) and is provided in Appendix A.1.

Lemma 4 *Suppose that the functions* f *and* g *of problem* (1) *satisfy Assumption* **A**.1*. If* $\lambda(\mathbf{x}) := \|\nabla f(\mathbf{x}) + \mathbf{v}\|^*_{\mathbf{x}} < 1$*, for some* $\mathbf{x} \in \mathrm{dom}(F)$ *and* $\mathbf{v} \in \partial g(\mathbf{x})$ *such that* $\nabla^2 f(\mathbf{x}) \succ 0$*, then the solution* $\mathbf{x}^*$ *of* (1) *exists and is unique.*

Since this problem is convex, the following optimality condition is necessary and sufficient:

$$\mathbf{0} \in \nabla f(\mathbf{x}^*) + \partial g(\mathbf{x}^*). \tag{11}$$

The solution $\mathbf{x}^*$ is called *strongly regular* if $\nabla^2 f(\mathbf{x}^*) \succ 0$. In this case, $\infty > \sigma^*_{\max} \geq \sigma^*_{\min} > 0$, where $\sigma^*_{\min}$ and $\sigma^*_{\max}$ are the smallest and the largest eigenvalue of $\nabla^2 f(\mathbf{x}^*)$, respectively.

Fixed-point characterization: Let $\mathbf{H} \in \mathbb{S}^n_+$. We define $S_{\mathbf{H}}(\mathbf{x}) := \mathbf{H}\mathbf{x} - \nabla f(\mathbf{x})$. Then, from (11), we have

$$S_{\mathbf{H}}(\mathbf{x}^*) \equiv \mathbf{H}\mathbf{x}^* - \nabla f(\mathbf{x}^*) \in \mathbf{H}\mathbf{x}^* + \partial g(\mathbf{x}^*).$$

By using the definition of $P^g_{\mathbf{H}}(\cdot)$ in (6), one can easily derive the fixed-point expression:

$$\mathbf{x}^* = P^g_{\mathbf{H}}\left(S_{\mathbf{H}}(\mathbf{x}^*)\right), \tag{12}$$

that is, $\mathbf{x}^*$ is the fixed-point of the mapping $R^g_{\mathbf{H}}(\cdot)$, where $R^g_{\mathbf{H}}(\cdot) := P^g_{\mathbf{H}}(S_{\mathbf{H}}(\cdot))$. The formula in (12) suggests that we can generate an iterative sequence based on the fixed-point principle, i.e., $\mathbf{x}^{k+1} := R^g_{\mathbf{H}}(\mathbf{x}^k)$ starting from $\mathbf{x}^0 \in \mathrm{dom}(F)$ for $k \geq 0$. Theoretically, under certain assumptions, one can ensure that the mapping $R^g_{\mathbf{H}}$ is contractive and the sequence generated by this scheme is convergent.

We note that if $g \equiv 0$ and $\mathbf{H} \in \mathbb{S}^n_{++}$, then $P^g_{\mathbf{H}}$ defined by (6) reduces to $P^g_{\mathbf{H}}(\cdot) = \mathbf{H}^{-1}(\cdot)$. Consequently, the fixed-point formula (12) becomes $\mathbf{x}^* = \mathbf{x}^* - \mathbf{H}^{-1}\nabla f(\mathbf{x}^*)$, which is equivalent to $\nabla f(\mathbf{x}^*) = 0$.

Our variable metric framework: Given a point $\mathbf{x}^k \in \mathrm{dom}(F)$ and a symmetric positive semidefinite matrix $\mathbf{H}_k$, we consider the function

$$Q(\mathbf{x}; \mathbf{x}^k, \mathbf{H}_k) := f(\mathbf{x}^k) + \nabla f(\mathbf{x}^k)^T(\mathbf{x} - \mathbf{x}^k) + \frac{1}{2}(\mathbf{x} - \mathbf{x}^k)^T\mathbf{H}_k(\mathbf{x} - \mathbf{x}^k), \tag{13}$$

for $\mathbf{x} \in \mathrm{dom}(F)$. The function $Q(\cdot; \mathbf{x}^k, \mathbf{H}_k)$ is—seemingly—a quadratic approximation of f around $\mathbf{x}^k$. Now, we study the following scheme to generate a sequence $\left\{\mathbf{x}^k\right\}_{k\geq 0}$:

$$\mathbf{x}^{k+1} := \mathbf{x}^k + \alpha_k \mathbf{d}^k, \tag{14}$$

where $\alpha_k \in (0, 1]$ is a step size and $\mathbf{d}^k$ is a search direction.

Let $\mathbf{s}^k$ be a solution of the following problem:

$$\mathbf{s}^k \in \mathcal{S}(\mathbf{x}^k, \mathbf{H}_k) := \mathop{\arg\min}_{\mathbf{x}\in\mathrm{dom}(F)} \left\{Q(\mathbf{x}; \mathbf{x}^k, \mathbf{H}_k) + g(\mathbf{x})\right\} = P^g_{\mathbf{H}_k}\left(\mathbf{H}_k\mathbf{x}^k - \nabla f(\mathbf{x}^k)\right). \tag{15}$$

Since we do not assume that $\mathbf{H}_k$ to be positive definite, the solution $\mathbf{s}^k$ may not exist. We require the following assumption:

Assumption A.2 *The subproblem* (15) *has at least one solution* $\mathbf{s}^k$*, i.e.,* $\mathcal{S}(\mathbf{x}^k, \mathbf{H}_k) \neq \emptyset$.

In particular, if $\mathbf{H}_k \in \mathbb{S}^n_{++}$, then the solution $\mathbf{s}^k$ of (15) exists and is unique, i.e., $\mathcal{S}(\mathbf{x}^k, \mathbf{H}_k) = \left\{\mathbf{s}^k\right\} \neq \emptyset$. Up to now, we have not required the uniqueness of $\mathbf{s}^k$. This assumption will be specified later in the next sections. Throughout this paper, we assume that both Assumptions **A**.1 and **A**.2 are satisfied without referring to them specifically.

Now, given $\mathbf{s}^k$, the direction $\mathbf{d}^k$ is computed as

$$\mathbf{d}^k := \mathbf{s}^k - \mathbf{x}^k. \tag{16}$$

If we define $\mathbf{G}_k := \mathbf{H}_k\mathbf{d}^k$, then $\mathbf{G}_k$ is called the *gradient mapping* of (1) (Nesterov, 2004), which behaves similarly as gradient vectors in non-composite minimization. Since problem (15) is solvable due to Assumption **A**.2, we can write its optimality condition as

$$\mathbf{0} \in \nabla f(\mathbf{x}^k) + \mathbf{H}_k(\mathbf{s}^k - \mathbf{x}^k) + \partial g(\mathbf{s}^k). \tag{17}$$

It is easy to see that if $\mathbf{d}^k = 0$, i.e., $\mathbf{s}^k \equiv \mathbf{x}^k$, then (17) reduces to $0 \in \nabla f(\mathbf{x}^k) + \partial g(\mathbf{x}^k)$, which is exactly (11). Hence, $\mathbf{x}^k$ is a solution of (1).

In the variable metric framework, depending on the choice of $\mathbf{H}_k$, the iteration scheme (14) leads to different methods for solving (1). For instance:

1. If $\mathbf{H}_k := \nabla^2 f(\mathbf{x}^k)$, then the method (14) is a *proximal-Newton* method.

2. If $\mathbf{H}_k$ is a symmetric positive definite matrix approximation of $\nabla^2 f(\mathbf{x}^k)$, then the method (14) is a *proximal-quasi Newton* method.

3. If $\mathbf{H}_k := L_k\mathbb{I}$, where L_k is, say, an approximation for the local Lipschitz constant of f and $\mathbb{I}$ is the identity matrix, then the method (14) is a *proximal-gradient* method.

Many of these above methods have been studied for (1) when $f \in \mathcal{F}_L$: cf., (Beck and Teboulle, 2009a; Becker and Fadili, 2012; Chouzenoux et al., 2013; Lee et al., 2012). Note however that, since the self-concordant part f of F is not (necessarily) globally Lipschitz continuously differentiable, these approaches are generally not applicable in theory.

Given the search direction $\mathbf{d}^k$ defined by (16), we define the following proximal-Newton decrement[4] λ_k and the weighted [semi-]norm β_k

$$\lambda_k := \|\mathbf{d}^k\|_{\mathbf{x}^k} = \Big((\mathbf{d}^k)^T\nabla^2 f(\mathbf{x}^k)\mathbf{d}^k\Big)^{1/2} \text{ and } \beta_k := \|\mathbf{d}^k\|_{\mathbf{H}_k}. \tag{18}$$

In the sequel, we study three different instances of the variable metric strategy in detail. Since we do not assume $\nabla^2 f(\mathbf{x}^k) \succ 0$, $\lambda_k = 0$ may not imply $\mathbf{d}^k = 0$.

Remark 5 *If $g \equiv 0$ and $\nabla^2 f(\mathbf{x}^k) \in \mathbb{S}^n_{++}$, then $\mathbf{d}^k = -\nabla^2 f(\mathbf{x}^k)^{-1}\nabla f(\mathbf{x}^k)$ is the standard Newton direction. In this case, λ_k defined by* (18) *reduces to $\lambda_k \equiv \|\nabla f(\mathbf{x}^k)\|^*_{\mathbf{x}^k}$, the Newton decrement defined in (Nesterov, 2004, Chapter 4). Moreover, we have $\lambda_k \equiv \lambda(\mathbf{x}^k)$, as defined in Lemma 4.*

3.1 A Proximal-Newton Method

If we choose $\mathbf{H}_k := \nabla^2 f(\mathbf{x}^k)$, then the method described in (14) is called the *proximal Newton* algorithm. For notational ease, we redefine $\mathbf{s}^k_n := \mathbf{s}^k$ and $\mathbf{d}^k_n := \mathbf{d}^k$, where the subscript n is used to distinguish proximal Newton related quantities from the other variable

4. This notion is borrowed from standard the Newton decrement defined in (Nesterov, 2004, Chapter 4).

metric strategies. Moreover, we use the shorthand notation $P^g_{\bar{\mathbf{x}}} := P^g_{\nabla^2 f(\bar{\mathbf{x}})}$, whenever $\bar{\mathbf{x}} \in \mathrm{dom}(f)$. Using (15) and (16), $\mathbf{s}^k_n$ and $\mathbf{d}^k_n$ are given by

$$\mathbf{s}^k_n := P^g_{\mathbf{x}^k}\left(\nabla^2 f(\mathbf{x}^k)\mathbf{x}^k - \nabla f(\mathbf{x}^k)\right), \quad \mathbf{d}^k_n := \mathbf{s}^k_n - \mathbf{x}^k. \tag{19}$$

Then, the proximal-Newton method generates a sequence $\left\{\mathbf{x}^k\right\}_{k\geq 0}$ starting from $\mathbf{x}^0 \in \mathrm{dom}(F)$ according to

$$\mathbf{x}^{k+1} := \mathbf{x}^k + \alpha_k \mathbf{d}^k_n, \tag{20}$$

where $\alpha_k \in (0, 1]$ is a step size. If $\alpha_k < 1$, then the iteration (20) is called the *damped proximal-Newton* iteration. If $\alpha_k = 1$, then it is called the *full-step proximal-Newton* iteration.

Global convergence: We first show that with an appropriate choice of the step-size $\alpha_k \in (0, 1]$, the iterative sequence $\left\{\mathbf{x}^k\right\}_{k\geq 0}$ generated by the damped-step proximal Newton scheme (20) is a decreasing sequence; i.e., $F(\mathbf{x}^{k+1}) \leq F(\mathbf{x}^k) - \omega(\sigma)$ whenever $\lambda_k \geq \sigma$, where $\sigma > 0$ is fixed. The following theorem provides an explicit formula for the step size α_k whose proof can be found in Appendix A.2.

Theorem 6 *If $\alpha_k := (1 + \lambda_k)^{-1} \in (0, 1]$, then the scheme in (20) generates $\mathbf{x}^{k+1}$ satisfies*

$$F(\mathbf{x}^{k+1}) \leq F(\mathbf{x}^k) - \omega(\lambda_k). \tag{21}$$

Moreover, the step α_k is optimal. The number of iterations to reach the point $\mathbf{x}^k$ such that $\lambda_k < \sigma$ for some $\sigma \in (0, 1)$ is $k_{\max} := \left\lfloor \frac{F(\mathbf{x}^0) - F(\mathbf{x}^*)}{\omega(\sigma)} \right\rfloor + 1$.

Local quadratic convergence rate: For any $\mathbf{x} \in \mathrm{dom}(f)$ such that $\nabla^2 f(\mathbf{x}) \succ 0$, we define the Dikin ellipsoid $\mathcal{W}^0(\mathbf{x}, r)$ as $\mathcal{W}^0(\mathbf{x}, r) := \left\{\mathbf{y} \in \mathrm{dom}(f) : \|\mathbf{y} - \mathbf{x}\|_{\mathbf{x}} < r\right\}$, see (Nesterov, 2004). We now establish the local quadratic convergence of the scheme (20). A complete proof of this theorem can be found in Appendix A.3.

Theorem 7 *Suppose that $\mathbf{x}^*$ is the unique solution of (1) and is strongly regular. Suppose further that $\nabla^2 f(\mathbf{x}) \succ 0$ for all $\mathbf{x} \in \mathcal{W}^0(\mathbf{x}^*, 1)$. Let $\left\{\mathbf{x}^k\right\}_{k\geq 0}$ be a sequence generated by the proximal Newton scheme (20) with $\alpha_k \in (0, 1]$. Then:*

a) *If $\alpha_k\lambda_k < 1 - \frac{1}{\sqrt{2}}$, then it holds that*

$$\lambda_{k+1} \leq \left(\frac{1 - \alpha_k + (2\alpha_k^2 - \alpha_k)\lambda_k}{1 - 4\alpha_k\lambda_k + 2\alpha_k^2\lambda_k^2}\right)\lambda_k. \tag{22}$$

b) *If the sequence $\left\{\mathbf{x}^k\right\}_{k\geq 0}$ is generated by the damped proximal-Newton scheme (20), starting from $\mathbf{x}^0$ such that $\lambda_0 \leq \bar{\sigma} := \sqrt{5} - 2 \approx 0.236068$ and $\alpha_k := (1 + \lambda_k)^{-1}$, then $\{\lambda_k\}_k$ locally converges to 0^+ at a quadratic rate.*

c) *Alternatively, if the sequence $\left\{\mathbf{x}^k\right\}_{k\geq 0}$ is generated by the full-step proximal-Newton scheme (20) starting from $\mathbf{x}^0$ such that $\lambda_0 \leq \bar{\sigma} := 0.25(5 - \sqrt{17}) \approx 0.219224$ and $\alpha_k = 1$, then $\{\lambda_k\}_k$ locally converges to 0^+ at a quadratic rate.*

Consequently, the sequence $\{\mathbf{x}^k\}_{k\geq 0}$ *also locally converges to* $\mathbf{x}^*$ *at a quadratic rate in both cases* b) *and* c)*, i.e.,* $\{\|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*}\}_{k\geq 0}$ *locally converges to* 0^+ *at a quadratic rate.*

A two-phase algorithm for solving (1)*:* Now, by the virtue of the above analysis, we can propose a two-phase proximal-Newton algorithm for solving (1). Initially, we perform the damped-step proximal-Newton iterations until we reach the quadratic convergence region (Phase 1). Then, we perform full-step proximal-Newton iterations, until we reach the desired accuracy (Phase 2). The pseudocode of the algorithm is presented in Algorithm 1.

Algorithm 1 (*Proximal-Newton algorithm*)

Inputs: $\mathbf{x}^0 \in \mathrm{dom}(F)$, tolerance $\varepsilon > 0$.

Initialization: Select a constant $\sigma \in (0, \frac{(5-\sqrt{17})}{4}]$, e.g., $\sigma := 0.2$.

for $k = 0$ **to** $K_{\max}$ **do**

1. Compute the proximal-Newton search direction $\mathbf{d}_n^k$ as in (19).
2. Compute $\lambda_k := \|\mathbf{d}_n^k\|_{\mathbf{x}^k}$.
3. **if** $\lambda_k > \sigma$ **then** $\mathbf{x}^{k+1} := \mathbf{x}^k + \alpha_k \mathbf{d}_n^k$, where $\alpha_k := (1 + \lambda_k)^{-1}$.
4. **elseif** $\lambda_k > \varepsilon$ **then** $\mathbf{x}^{k+1} := \mathbf{x}^k + \mathbf{d}_n^k$.
5. **else** terminate.

end for

The radius σ of the quadratic convergence region in Algorithm 1 can be fixed at any value in $(0, \bar{\sigma}]$, e.g., at its upper bound $\bar{\sigma}$. An upper bound $K_{\max}$ of the iterations can also be specified, if necessary. The computational bottleneck in Algorithm 1 is typically incurred Step 1 in Phase 1 and Phase 2, where we need to solve the subproblem (15) to obtain a search direction $\mathbf{d}_n^k$. When problem (15) is strongly convex, i.e., $\nabla^2 f(\mathbf{x}^k) \in \mathbb{S}_{++}^n$, one can apply first order methods to efficiently solve this problem with a linear convergence rate (see, e.g., Beck and Teboulle 2009a; Nesterov 2004, 2007) and make use of a *warm-start* strategy by employing the information of the previous iterations.

Remark 8 *From Remark 3 we see that if* $\nabla f(\mathbf{x}^k) \succeq 0$*, then* $\lambda^k = 0$ *may not imply* $\mathbf{d}^k = 0$*. Therefore, we can add an auxiliary stopping criterion* $\beta_k := \|\mathbf{d}^k\|_2 \leq \varepsilon$ *to Algorithm 1 so that we can avoid the termination of Algorithm 1 at a non-optimal point* $\mathbf{x}^k$*.*

*Iteration-complexity analysis.*The choice of σ in Algorithm 1 can trade-off the number of iterations between the damped-step and full-step iterations. If we fix $\sigma = 0.2$, then the complexity of the full-step Newton phase becomes $\mathcal{O}\left(\ln\ln\left(\frac{0.28}{\varepsilon}\right)\right)$. The following theorem summarizes the complexity of the proposed algorithm.

Theorem 9 *The maximum number of iterations required in Algorithm 1 does not exceed* $K_{\max} := \left\lfloor \frac{F(\mathbf{x}^0) - F(\mathbf{x}^*)}{0.017} \right\rfloor + \left\lfloor 1.5\left(\ln\ln\left(\frac{0.28}{\varepsilon}\right)\right)\right\rfloor + 2$ *provided that* $\sigma = 0.2$ *to obtain* $\lambda_k \leq \varepsilon$*. Consequently,* $\|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*} \leq 2\varepsilon$*, where* $\mathbf{x}^*$ *is the unique solution of* (1)*.*

Proof Let $\sigma = 0.2$. From the estimate (22) of Theorem 7 and $\alpha_{k-1} = 1$ we have $\lambda_k \leq (1 - 4\lambda_{k-1} + 2\lambda_{k-1}^2)^{-1}\lambda_{k-1}^2$ for $k \geq 1$. Since $\lambda_0 \leq \sigma$, by induction, we can easily show

that $\lambda_k \le (1 - 4\sigma + 2\sigma^2)^{-1}\lambda_{k-1}^2 \le c\lambda_{k-1}^2$, where $c := 3.57$. This implies $\lambda_k \le c^{2^k-1}\lambda_0^{2^k} \le c^{2^k-1}\sigma^{2^k}$. The stopping criterion $\lambda_k \le \varepsilon$ in Algorithm 1 is ensured if $(c\sigma)^{2^k} \le c\varepsilon$. Since $c\sigma \approx 0.71 < 1$, the last condition leads to $k \ge (\ln 2)^{-1}\ln\left(\frac{-\ln(c\sigma)}{-\ln(c\varepsilon)}\right)$. By using $c = 3.57$, $\sigma = 0.2$ and the fact that $\ln(2)^{-1} < 1.5$, we can show that the last requirement is fulfilled if $k \ge \left\lfloor 1.5\left(\ln\ln\left(\frac{0.28}{\varepsilon}\right)\right)\right\rfloor + 1$. Now, combining the last conclusion and Theorem 6 with noting that $\omega(\sigma) > 0.017$ we obtain $K_{\max}$ as in Theorem 9.

Finally, we prove $\|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*} \le 2\varepsilon$. Indeed, we have $r_k := \|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*} \le \frac{\|\mathbf{x}^{k+1}-\mathbf{x}^k\|_{\mathbf{x}^k}}{1-\|\mathbf{x}^k-\mathbf{x}^*\|_{\mathbf{x}^*}} + \|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^*} = \frac{\lambda_k}{1-r_k} + r_{k+1}$, whenever $r_k < 1$. Next, using (84) with $\alpha_k = 1$, we have $r_{k+1} \le \frac{(3-r_k)r_k^2}{1-4r_k+2r_k^2}$. Combining these inequalities, we obtain $\frac{(1-r_k)(1-7r_k+3r_k^2)r_k}{1-4r_k+2r_k^2} \le \lambda_k \le \varepsilon$. Since the function $s(r) := \frac{(1-r)(1-7r+3r^2)r}{1-4r+2r^2}$ attains a maximum at $r^* \approx 0.08763$ and it is increasing on $[0, r^*]$. Moreover, $\frac{(1-r_k)(1-7r_k+3r_k^2)}{1-4r_k+2r_k^2} \ge 0.5$ for $r_k \in [0, r^*]$, which leas to $0.5r_k \le \frac{(1-r_k)(1-7r_k+3r_k^2)r_k}{1-4r_k+2r_k^2} \le \varepsilon$. Hence, $r_k \le 2\varepsilon$ provided that $r_k \le r_0 \le r^* \approx 0.08763$. ■

Remark 10 *When $g \equiv 0$, we can modify the proof of estimate* (22) *to obtain a tighter bound $\lambda_{k+1} \le \frac{\lambda_k^2}{(1-\lambda_k)^2}$. This estimate is exactly (Nesterov, 2004), which implies that the radius of the quadratic convergence region is $\bar{\sigma} := (3 - \sqrt{5})/2$.*

A modification of the proximal-Newton method: In Algorithm 1, if we remove Step 4 and replace analytic step-size selection calculation in Step 3 with a backtracking line-search, then we reach the proximal Newton method of (Lee et al., 2012). Hence, this approach *in practice* might lead to reduced overall computation since our step-size α_k is selected optimally with respect to the worst case problem structures as opposed to the particular instance of the problem. Since the backtracking approach always starts with the full-step, we also do not need to know whether we are within the quadratic convergence region. Moreover, the cost of evaluating the objective at the full-step in certain applications may not be significantly worse than the cost of calculating α_k or may be dominated by the cost of calculating the Newton direction.

In stark contrast to backtracking, our new theory behooves us to propose a new forward line-search procedure as illustrated by Figure 2. The idea is quite simple: we start with the

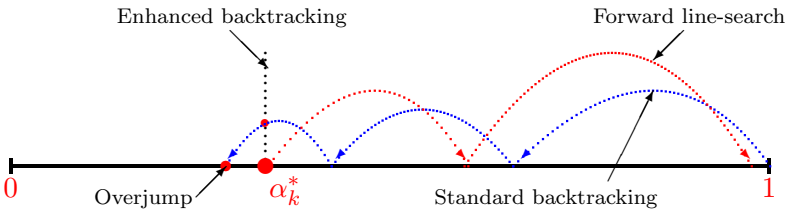


Figure 2: Illustration of step-size selection procedures.

"optimal" step-size α_k and increase it towards full-step with a stopping condition based on the objective evaluations. Interestingly, when we analytically calculate the step, we also have

access to the side information on whether or not we are within the quadratic convergence region, and hence, we can automatically switch to Step 4 in Algorithm 1. Alternatively, calculation of the analytic step-size can enhance backtracking since the knowledge of α_k reduces the backtracking range from $(0, 1]$ to $(\alpha_k, 1]$ with the side-information as to when to automatically take the full-step without function evaluation.

3.2 A Proximal Quasi-Newton Scheme

Even if the function f is self-concordant, the numerical evaluation of $\nabla^2 f(\mathbf{x})$ can be expensive in many applications (e.g., $f(\mathbf{x}) := \sum_{j=1}^{p} f_j(\mathbf{A}_j\mathbf{x})$, with $p \gg n$). Hence, it is interesting to study proximal quasi-Newton method for solving (1). Our interest in the quasi-Newton methods in this paper is for completeness; we do not provide any algorithmic details or implementations on our quasi-Newton variant.

To this end, we need a symmetric positive definite matrix $\mathbf{H}_k$ that approximates $\nabla^2 f(\mathbf{x}^k)$ at the iteration k. As a result, our main assumption here is that matrix $\mathbf{H}_{k+1}$ at the next iteration $k+1$ satisfies the *secant equation*:

$$\mathbf{H}_{k+1}(\mathbf{x}^{k+1} - \mathbf{x}^k) = \nabla f(\mathbf{x}^{k+1}) - \nabla f(\mathbf{x}^k). \tag{23}$$

For instance, it is well-known that the sequence of matrices $\{\mathbf{H}_k\}_{k\geq 0}$ updated by the following BFGS formula satisfies the secant equation (23) (Nocedal and Wright, 2006):

$$\mathbf{H}_{k+1} := \mathbf{H}_k + \frac{1}{(\mathbf{y}^k)^T\mathbf{z}^k}\mathbf{y}^k(\mathbf{y}^k)^T - \frac{1}{(\mathbf{z}^k)^T\mathbf{H}_k\mathbf{z}^k}\mathbf{H}_k\mathbf{z}^k(\mathbf{H}_k\mathbf{z}^k)^T, \tag{24}$$

where $\mathbf{z}^k := \mathbf{x}^{k+1} - \mathbf{x}^k$ and $\mathbf{y}^k := \nabla f(\mathbf{x}^{k+1}) - \nabla f(\mathbf{x}^k)$. Other methods for updating matrix $\mathbf{H}_k$ can be found in (Nocedal and Wright, 2006), which are not listed here.

In this subsection, we only analyze the full-step proximal quasi-Newton scheme based on the BFGS updates. The global convergence characterization of the BFGS quasi-Newton method can be obtained using our analysis in the next subsection. To this end, we have the following update equation, where the subscript q is used to distinguish the quasi-Newton method:

$$\mathbf{x}^{k+1} := \mathbf{x}^k + \mathbf{d}_q^k. \tag{25}$$

Here we use $\mathbf{d}_q^k$ to stand for the proximal quasi-Newton search direction.

Under certain assumptions, one can prove that the sequence $\left\{\mathbf{x}^k\right\}_{k\geq 0}$ generated by (25) converges to $\mathbf{x}^*$ the unique solution of (1). One of the common assumptions used in quasi-Newton methods is the Dennis-Moré condition, see (Dennis and Moré, 1974). Adopting the Dennis-Moré criterion, we impose the following condition in our context:

$$\lim_{k\to\infty} \frac{\left\|\left[\mathbf{H}_k - \nabla^2 f(\mathbf{x}^*)\right](\mathbf{x}^{k+1} - \mathbf{x}^k)\right\|_{\mathbf{x}^*}^*}{\|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^*}} = 0. \tag{26}$$

The Dennis-Moré condition becomes standard in smooth optimization. Examples can be found, e.g., in (Byrd and Nocedal, 1989; Nocedal and Wright, 2006). Now, we establish the superlinear convergence of the sequence $\left\{\mathbf{x}^k\right\}_{k\geq 0}$ generated by (25) as follows.

Theorem 11 *Assume that* $\mathbf{x}^*$ *is the unique solution of* (1) *and is strongly regular. Let matrix* $\mathbf{H}_k$ *maintains the secant equation* (23) *and let* $\{\mathbf{x}^k\}_{k\geq 0}$ *be a sequence generated by scheme* (25)*. Then the following statements hold:*

(a) Suppose, in addition, that the sequence of matrices $\{\mathbf{H}_k\}_{k\geq 0}$ *satisfies the Dennis-Moré condition* (26) *for sufficiently large* k*. Then the sequence* $\{\mathbf{x}^k\}_{k\geq 0}$ *converges to the solution* $\mathbf{x}^*$ *of* (1) *at a superlinear rate provided that* $\|\mathbf{x}^0 - \mathbf{x}^*\|_{\mathbf{x}^*} < 1$*.*

(b) Suppose that a matrix $\mathbf{H}_0 \succ 0$ *is chosen. Then* $(\mathbf{y}^k)^T\mathbf{z}^k > 0$ *for all* $k \geq 0$ *and hence the sequence* $\{\mathbf{H}_k\}_{k\geq 0}$ *generated by* (24) *is symmetric positive definite and satisfies the secant equation* (23)*. Moreover, if the sequence* $\{\mathbf{x}^k\}_{k\geq 0}$ *generated by* (25) *satisfies* $\sum_{k=0}^{\infty}\|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*} < +\infty$*, then this sequence converges to* $\mathbf{x}^*$ *at a superlinear rate.*

The proof of this theorem can be found in Appendix A.3. We note that if the sequence $\{\mathbf{x}^k\}_{k\geq 0}$ locally converges to $\mathbf{x}^*$ at a linear rate w.r.t. the local norm at $\mathbf{x}^*$, i.e. $\|\mathbf{x}^{k+1} - \mathbf{x}^*\|_{\mathbf{x}^*} \leq \kappa\|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*}$ for some $\kappa \in (0,1)$ and $k \geq 0$, then the condition $\sum_{k=0}^{\infty}\|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*} < +\infty$ automatically holds. From (26) we also observe that the matrix $\mathbf{H}_k$ is required to well approximate $\nabla^2 f(\mathbf{x}^*)$ along the direction $\mathbf{d}_q^k$, which is not in the whole space.

3.3 A Proximal-Gradient Method

If we choose matrix $\mathbf{H}_k := \mathbf{D}_k$, where $\mathbf{D}_k$ is a positive diagonal matrix, then the iterative scheme (14) is called the *proximal-gradient* scheme. In this case, we can write (14) as

$$\mathbf{x}^{k+1} := \mathbf{x}^k + \alpha_k\mathbf{d}_g^k = (1-\alpha_k)\mathbf{x}^k + \alpha_k\mathbf{s}_g^k, \tag{27}$$

where $\alpha_k \in (0,1]$ is an appropriate step size, $\mathbf{d}_g^k$ is the proximal-gradient search direction and $\mathbf{s}_g^k \equiv \mathbf{s}^k$ as in (15).

The following lemma shows how we can choose the step size α_k corresponding to $\mathbf{D}_k$ such that we obtain a descent direction in the proximal-gradient scheme (27). The proof of this lemma can be found in Appendix A.2.

Lemma 12 *Let* $\{\mathbf{x}^k\}_{k\geq 0}$ *be a sequence generated by* (27)*. Suppose that the matrix* $\mathbf{D}_k \succ 0$ *is chosen such that the step size* α_k *satisfies* $\alpha_k := \frac{\beta_k^2}{\lambda_k(\lambda_k+\beta_k^2)} \in (0,1]$ *(see below), where* $\beta_k := \|\mathbf{d}_g^k\|_{\mathbf{D}_k}$ *and* $\lambda_k := \|\mathbf{d}_g^k\|_{\mathbf{x}^k}$*. Then* $\{\mathbf{x}^k\}_{k\geq 0} \subset \mathrm{dom}(F)$ *and the following estimate holds*

$$F(\mathbf{x}^{k+1}) \leq F(\mathbf{x}^k) - \omega\left(\beta_k^2/\lambda_k\right), \tag{28}$$

where $\omega(\tau) := \tau - \ln(1+\tau) \geq 0$*.*

From Lemma 12, we observe that $\alpha_k \leq 1$ if $\frac{\lambda_k^2}{\beta_k^2} + \lambda_k \geq 1$. It is obvious that if $\lambda_k \geq 1$ then the last condition is automatically satisfied. We only consider the case $\lambda_k < 1$. In fact, since $\lambda_k \geq 0$, we relax actually the condition $\frac{\lambda_k^2}{\beta_k^2} + \lambda_k \geq 1$ to a simpler condition $\lambda_k \geq \beta_k$.

Algorithm 2 (*Proximal-gradient method*)

Inputs: $\mathbf{x}^0 \in \mathrm{dom}(F)$, tolerance $\varepsilon > 0$.

for $k = 0$ **to** $k_{\max}$ **do**

1. Choose an appropriate $\mathbf{D}_k \succ 0$ based on (30).
2. Compute $\mathbf{d}_g^k := \mathcal{P}_{\mathbf{D}_k}^g\left(\mathbf{D}_k\mathbf{x}^k - \nabla f(\mathbf{x}^k)\right) - \mathbf{x}^k$ due to (15).
3. Compute $\beta_k := \|\mathbf{d}_g^k\|_{\mathbf{D}_k}$ and $\lambda_k := \|\mathbf{d}_g^k\|_{\mathbf{x}^k}$.
4. If $e_k := \|\mathbf{d}_g^k\|_2 \leq \varepsilon$ then terminate.
5. Update $\mathbf{x}^{k+1} := \mathbf{x}^k + \alpha_k\mathbf{d}_g^k$, where $\alpha_k := \frac{\beta_k^2}{\lambda_k(\lambda_k+\beta_k^2)} \in (0, 1]$.

end for

We now study the case $\mathbf{D}_k := L_k\mathbb{I}$, where $L_k \geq \underline{L} > 0$ is a positive constant and $\mathbb{I}$ is the identity matrix with dimensions apparent from the context. Hence, $\beta_k^2 = L_k\|\mathbf{d}_g^k\|_2^2$ and

$$\frac{\lambda_k^2}{\beta_k^2} = \frac{(\mathbf{d}_g^k)^T\nabla^2 f(\mathbf{x}^k)\mathbf{d}_g^k}{L_k\|\mathbf{d}_g^k\|_2^2}.$$

However, since

$$\sigma_{\min}(\nabla^2 f(\mathbf{x}^k)) \leq \sigma^k := \frac{(\mathbf{d}_g^k)^T\nabla^2 f(\mathbf{x}^k)\mathbf{d}_g^k}{\|\mathbf{d}_g^k\|_2^2} \leq \sigma_{\max}(\nabla^2 f(\mathbf{x}^k)), \tag{29}$$

the condition $\lambda_k \geq \beta_k$ is equivalent to

$$L_k \leq \sigma_k, \tag{30}$$

where $\sigma_{\min}^k := \sigma_{\min}(\nabla^2 f(\mathbf{x}^k))$ and $\sigma_{\max}^k := \sigma_{\max}(\nabla^2 f(\mathbf{x}^k))$ are the smallest and largest eigenvalue of $\nabla^2 f(\mathbf{x}^k)$, respectively. Under the assumption that $\mathrm{dom}(f)$ contains no straight-line, then we have the Hessian $\nabla^2 f(\mathbf{x}^k) \succ 0$ by (Nesterov, 2004, Theorem 4.1.3), which implies that $\sigma_{\min}^k > 0$. Therefore, in the worst-case, we can choose $L_k := \sigma_{\min}^k$. However, this lower bound may be too conservative. In practice, we can apply a *bisection procedure* to meet the condition (30). It is not difficult to prove via contradiction that the number of bisection steps is upper bounded by a constant.

We note that if g is separable, i.e., $g(\mathbf{x}) := \sum_{i=1}^n g_i(\mathbf{x}_i)$ (e.g., $g(\mathbf{x}) := \rho\,\|\mathbf{x}\|_1$), then we can compute $\mathbf{s}_{\mathbf{D}_k}^k$ in (15) in a component-wise fashion as

$$(\mathbf{s}_{L_k}^k)_i := \mathcal{P}_{\tau_i^k}^{g_i}\left(\mathbf{x}_i^k - \tau_i^k(\nabla f(\mathbf{x}^k))_i\right),\ i = 1, \ldots, n, \tag{31}$$

where $\tau_i^k := 1/(\mathbf{D}_k)_{ii}$ and $\mathcal{P}_{\tau_i}^{g_i}(\cdot)$ is the proximity operator of g_i function, with parameter τ_i. The computation of λ_k only requires one matrix-vector multiplication and one vector inner-product; but it can be reduced by exploiting concrete structure of the smooth part f.

Based on Lemma 12, we describe the proximal-gradient scheme (27) in Algorithm 2. The main computation cost of Algorithm 2 is incurred at Step 2 and in calculating λ_k. If g is separable, then the computation of Step 2 can be done in a *closed form*. One main

step of Algorithm 2 is Step 2, which depends on the cost of prox-operator $\mathcal{P}^g_{\mathbf{D}_k}$. In practice, $\mathbf{D}_k$ is determined by a bisection procedure whenever $\lambda_k < 1$, which requires additional computational cost. If we choose $D_k := L_k\mathbb{I}$, then in order to fulfill (30), we can perform a back-tracking line search procedure on L_k. This line search procedure does not require the evaluations of the objective function. We modify Steps 1-3 of Algorithm 2 as

1. Initialize $L_k := L_k^0 > 0$, e.g., by a Barzilai-Borwein step.

2. Compute $\mathbf{d}_g^k := \mathcal{P}^g_{L_k\mathbf{I}_k}\left(L_k\mathbf{x}^k - \nabla f(\mathbf{x}^k)\right) - \mathbf{x}^k$ due to (15).

3a. Compute $\beta_k := \|\mathbf{d}_g^k\|_{L_k\mathbf{I}}$ and $\lambda_k := \|\mathbf{d}_g^k\|_{\mathbf{x}^k}$.

3b. If $\lambda_k^2/\beta_k^2 + \lambda_k < 1$, then set $L_k := L_k/2$ and go back to Step 2.

We note that computing λ_k at Step 3 does not need to form the full Hessian $\nabla^2 f(\mathbf{x}^k)$, it only requires a directional derivative, which is relatively cheap in applications (Nocedal and Wright, 2006, Chapter 7).

Global and local convergence. The global and local convergence of Algorithm 2 is stated in the following theorems, whose proof can be found in Appendix A.2.

Theorem 13 *Assume that there exists $\underline{L} > 0$ such that $\mathbf{D}_k \succeq \underline{L}\mathbb{I}$ for $k \geq 0$, and the solution $\mathbf{x}^*$ of (1) is unique. Let the sublevel set*

$$\mathcal{L}_F(F(\mathbf{x}^0)) := \left\{\mathbf{x} \in \mathrm{dom}(F) \mid F(\mathbf{x}) \leq F(\mathbf{x}^0)\right\}$$

be bounded. Then, the sequence $\left\{\mathbf{x}^k\right\}_{k\geq 0}$, generated by Algorithm 2, converges to the unique solution $\mathbf{x}^*$ of (1).

Theorem 14 *Assume that $\mathbf{x}^*$ is the unique solution of (1) and is strongly regular. Let $\left\{\mathbf{x}^k\right\}_{k\geq 0}$ be the sequence generated by Algorithm 2. Then, for k sufficiently large, if*

$$\frac{\left\|[\mathbf{D}_k - \nabla^2 f(\mathbf{x}^*)]\mathbf{d}_g^k\right\|^*_{\mathbf{x}^*}}{\|\mathbf{d}_g^k\|_{\mathbf{x}^*}} < \frac{1}{2}, \tag{32}$$

then $\left\{\mathbf{x}^k\right\}_{k\geq 0}$ locally converges to $\mathbf{x}^*$ at a linear rate. In particular, if $\mathbf{D}_k := L_k\mathbb{I}$ and $\gamma_* := \max\left\{\left|1 - \frac{L_k}{\sigma^*_{\min}}\right|, \left|1 - \frac{L_k}{\sigma^*_{\max}}\right|\right\} < \frac{1}{2}$, then the condition (32) holds.

We note that $\mathbf{x}^*$ is *unknown*; thus, evaluating γ_* a priori is infeasible in reality. In implementation, one can choose an appropriate value $L_k \geq \underline{L} > 0$ and then adaptively update L_k based on the knowledge of the eigenvalues of $\nabla^2 f(\mathbf{x}^k)$ near to the solution $\mathbf{x}^*$. The condition (32) can be expressed as $(\mathbf{d}_g^k)^T[L_k^2\nabla^2 f(\mathbf{x}^*)^{-1} + \nabla^2 f(\mathbf{x}^*) - 2L_k\mathbb{I}]\mathbf{d}_g^k \leq (1/4)\|\mathbf{d}_g^k\|^2_{\mathbf{x}^*}$, which leads to

$$(3/4)\|\mathbf{d}_g^k\|^2_{\mathbf{x}^*} + L^2[\|\mathbf{d}_g^k\|^*_{\mathbf{x}^*}]^2 < 2L_k\|\mathbf{d}_g^k\|_2^2. \tag{33}$$

We note that to find L_k such that (33) holds, we require $\|\mathbf{d}_g^k\|^*_{\mathbf{x}^*}\|\mathbf{d}_g^k\|_{\mathbf{x}^*} < \sqrt{\frac{4}{3}}\|\mathbf{d}_g^k\|_2^2$. If the last condition in Theorem 14 is satisfied then the condition (33) also holds. While the

last condition in Theorem 14 seems too imposing, we claim that, for most f and g, we only require (33) to be satisfied (see also the empirical evidence in Subsection 5.2.1). The condition (32) (or (33)) can be referred to as a **restricted** approximation gap between $\mathbf{D}_k$ and the true Hessian $\nabla^2 f(\mathbf{x}^*)$ along the direction $\mathbf{d}_g^k$ for k sufficiently large. For instance, when g is based on the ℓ_1-norm/the nuclear norm, the search direction $\mathbf{d}_g^k$ have at most twice the sparsity/rank of $\mathbf{x}^*$ near the convergence region.

Remark 15 *From the scheme* (27) *we observe that the step size* $\alpha_k < 1$ *may not preserve some of the desiderata on* $\mathbf{x}^{k+1}$ *due to the closed form solution of the prox-operator* $\mathcal{P}^g_{\mathbf{D}_k}$. *For instance, when* g *is based on the* ℓ_1*-norm,* $\alpha_k < 1$, *might increase the sparsity level of the solution as opposed to monotonically increasing it. However, in practice, the numerical values of* α_k *are often* 1 *near the convergence, which maintain properties, such as sparsity, low-rankedness, etc.*

Global convergence rate: In proximal gradient methods, proving global convergence rate guarantees requires a global constant to be known *a priori*—such as the Lipschitz constant. However such an assumption does not apply for the class of just self-concordant functions that we consider in this paper. We only characterize the following property in an ergodic sense. Let $\{\mathbf{x}^k\}_{k\geq 0}$ be the sequence generated by (2). We define

$$\bar{\mathbf{x}}^k := S_k^{-1} \sum_{j=0}^{k} \alpha_j \mathbf{x}^j, \quad \text{where} \quad S_k := \sum_{j=0}^{k} \alpha_j > 0. \tag{34}$$

Then we can show that $F(\bar{\mathbf{x}}^k) - F^* \leq \frac{\bar{L}_k}{2S_k} \left\|\mathbf{x}^0 - \mathbf{x}^*\right\|_2^2$, where $\bar{L}_k := \max_{0\leq j\leq k} L_j$. If $\alpha_j \geq \underline{\alpha} > 0$ for $0 \leq j \leq k$, then $S_k \geq \underline{\alpha}(k+1)$, which leads to $F(\bar{\mathbf{x}}^k) - F^* \leq \frac{\bar{L}_k}{2(k+1)\underline{\alpha}} \left\|\mathbf{x}^0 - \mathbf{x}^*\right\|_2^2$. The proof of this statement can be found in (Tran-Dinh et al., 2014b), which we omit here.

A modification of the proximal-gradient method: If the point $\mathbf{s}_g^k$ generated by (15) belongs to $\mathrm{dom}(F)$, then $F(\mathbf{s}_g^k) < +\infty$. Similarly to the definition of $\mathbf{x}^{k+1}$ in (27), we can define a new trial point:

$$\hat{\mathbf{x}}^k := (1-\alpha_k)\mathbf{x}^k + \alpha_k \mathbf{s}_g^k. \tag{35}$$

If $F(\mathbf{s}_g^k) \leq F(\mathbf{x}^k)$, then, by the convexity of F, it is easy to show that:

$$F(\hat{\mathbf{x}}^k) = F\big((1-\alpha_k)\mathbf{x}^k + \alpha_k \mathbf{s}_g^k\big) \leq (1-\alpha_k)F(\mathbf{x}^k) + \alpha_k F(\mathbf{s}_g^k) \overset{F(\mathbf{s}_g^k)\leq F(\mathbf{x}^k)}{\leq} F(\mathbf{x}^k).$$

In this case, based on the function values $F(\mathbf{s}_g^k)$, $F(\hat{\mathbf{x}}^k)$ and $F(\mathbf{x}^k)$ we can eventually choose the next iteration $\mathbf{x}^{k+1}$ as follows:

$$\mathbf{x}^{k+1} := \begin{cases} \mathbf{s}_g^k & \text{if } \mathbf{s}^k \in \mathrm{dom}(F) \text{ and } F(\mathbf{s}_g^k) < F(\hat{\mathbf{x}}^k) \quad \text{(Case 1)}, \\ \hat{\mathbf{x}}^k & \text{otherwise} \qquad\qquad\qquad\qquad\quad \text{(Case 2)}. \end{cases} \tag{36}$$

The idea of this *greedy* modification is illustrated in Figure 3. We note that here we need to check $\mathbf{s}_g^k \in \mathrm{dom}(F)$ such that $F(\mathbf{s}_g^k) < F(\mathbf{x}^k)$ and additional function evaluations $F(\mathbf{s}_g^k)$ and $F(\hat{\mathbf{x}}^k)$. However, careful implementations can recycle quantities that enable us to evaluate the objective at $\mathbf{s}_g^k$ and at $\mathbf{x}^{k+1}$ with very little overhead over the calculation of α_k (see Section 4). By using (36), we can specify a modified proximal gradient algorithm for solving (1), whose details we omit here since it is quite similar to Algorithm 2.

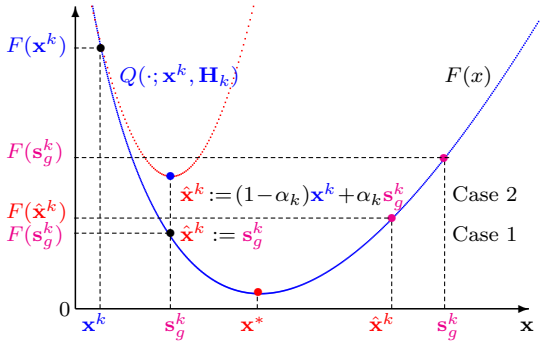


Figure 3: Illustration of the modified proximal-gradient method

4. Concrete Instances of our Optimization Framework

We illustrate three instances of our framework for some of the applications described in Section 1. For concreteness, we describe only the first and second order methods. Quasi-Newton methods based on (L-)BFGS updates or other adaptive variable metrics can be similarly derived in a straightforward fashion.

4.1 Graphical Model Selection

We customize our optimization framework to solve the graph selection problem (2). For notational convenience, we maintain a matrix variable $\boldsymbol{\Theta}$ instead of vectorizing it. We observe that $f(\boldsymbol{\Theta}) := -\log(\det(\boldsymbol{\Theta})) + \mathrm{tr}(\hat{\boldsymbol{\Sigma}}\boldsymbol{\Theta})$ is a standard self-concordant function, while $g(\boldsymbol{\Theta}) := \rho\,\|\mathbf{vec}(\boldsymbol{\Theta})\|_1$ is convex and nonsmooth. The gradient and the Hessian of f can be computed explicitly as $\nabla f(\boldsymbol{\Theta}) := \hat{\boldsymbol{\Sigma}} - \boldsymbol{\Theta}^{-1}$ and $\nabla^2 f(\boldsymbol{\Theta}) := \boldsymbol{\Theta}^{-1} \otimes \boldsymbol{\Theta}^{-1}$, respectively. Next, we formulate our proposed framework to construct two algorithmic variants for (2).

4.1.1 Dual Proximal-Newton Algorithm

We consider a second order algorithm via a dual solution approach for (15). This approach is first introduced in our earlier work (Tran-Dinh et al., 2013a), which did not consider the new modifications we propose in Section 3.1.

We begin by deriving the following dual formulation of the convex subproblem (15). Let $\mathbf{p}_k := \nabla f(\mathbf{x}^k)$, the convex subproblem (15) can then be written equivalently as

$$\min_{\mathbf{x}\in\mathbb{R}^n}\Big\{(1/2)\mathbf{x}^T\mathbf{H}_k\mathbf{x} + (\mathbf{p}_k - \mathbf{H}_k\mathbf{x}^k)^T\mathbf{x} + g(\mathbf{x})\Big\}. \tag{37}$$

By using the min-max principle, we can write (37) as

$$\max_{\mathbf{u}\in\mathbb{R}^n}\min_{\mathbf{x}\in\mathbb{R}^n}\Big\{(1/2)\mathbf{x}^T\mathbf{H}_k\mathbf{x} + (\mathbf{p}_k - \mathbf{H}_k\mathbf{x}^k)^T\mathbf{x} + \mathbf{u}^T\mathbf{x} - g^*(\mathbf{u})\Big\}, \tag{38}$$

where g^* is the Fenchel conjugate function of g, i.e., $g^*(\mathbf{u}) := \sup_{\mathbf{x}}\big\{\mathbf{u}^T\mathbf{x} - g(\mathbf{x})\big\}$. Solving the inner minimization in (38) we obtain

$$\min_{\mathbf{u}\in\mathbb{R}^n}\left\{(1/2)\mathbf{u}^T\mathbf{H}_k^{-1}\mathbf{u} + \tilde{\mathbf{p}}_k^T\mathbf{u} + g^*(\mathbf{u})\right\}, \tag{39}$$

where $\tilde{\mathbf{p}}_k := \mathbf{H}_k^{-1}\mathbf{p}_k - \mathbf{x}^k$. Note that the objective function $\varphi(\mathbf{u}) := g^*(\mathbf{u}) + (1/2)\mathbf{u}^T\mathbf{H}_k^{-1}\mathbf{u} + \tilde{\mathbf{p}}_k^T\mathbf{u}$ of (39) is strongly convex, one can apply the fast projected gradient methods with a linear convergence rate for solving this problem, see (Nesterov, 2007; Beck and Teboulle, 2009a).

In order to recover the solution of the primal subproblem (15), we note that the solution of the parametric minimization problem in (38) is given by $\mathbf{x}^*(\mathbf{u}) := \mathbf{x}^k - \mathbf{H}_k^{-1}(\mathbf{p}_k + \mathbf{u})$. Let $\mathbf{u}^*_{\mathbf{x}^k}$ be the optimal solution of (39). We can recover the primal proximal-Newton search direction $\mathbf{d}^k$ defined in (16) as

$$\mathbf{d}_n^k = -\nabla^2 f(\mathbf{x}^k)^{-1}\big(\nabla f(\mathbf{x}^k) + \mathbf{u}^*_{\mathbf{x}^k}\big). \tag{40}$$

To compute the quantity λ_k defined by (18) in Algorithm 1, we use (40) such that:

$$\lambda_k = \|\mathbf{d}_n^k\|_{\mathbf{x}^k} = \|\nabla f(\mathbf{x}^k) + \mathbf{u}^*_{\mathbf{x}^k}\|^*_{\mathbf{x}^k}. \tag{41}$$

Note that computing λ_k by (41) requires the inverse of the Hessian matrix $\nabla^2 f(\mathbf{x}^k)$.

Surprisingly, this dual approach allows us to avoid matrix inversion as well as Cholesky decomposition in computing the gradient $\nabla f(\mathbf{\Theta}_i)$ and the Hessian $\nabla^2 f(\mathbf{\Theta}_i)$ of f in graph selection. An alternative is of course to solve (15) in its primal form. Though, in such case, we need to compute $\mathbf{\Theta}_i^{-1}$ at each iteration i (say, via Cholesky decompositions).

The dual subproblem (39) becomes as

$$\mathbf{U}^* = \underset{\|\mathbf{vec}(\mathbf{U})\|_\infty \leq 1}{\arg\min} \Big\{(1/2)\mathrm{tr}((\mathbf{\Theta}_i\mathbf{U})^2) + \mathrm{tr}(\widetilde{\mathbf{Q}}\mathbf{U})\Big\}, \tag{42}$$

for the graph selection, where $\widetilde{\mathbf{Q}} := \rho^{-1}[\mathbf{\Theta}_i\widehat{\mathbf{\Sigma}}\mathbf{\Theta}_i - 2\mathbf{\Theta}_i]$. Given the dual solution $\mathbf{U}^*$ of (42), the primal proximal-Newton search direction (i.e. the solution of (15)) is computed as

$$\mathbf{\Delta}_i := -\big((\mathbf{\Theta}_i\widehat{\mathbf{\Sigma}} - \mathbb{I})\mathbf{\Theta}_i + \rho\mathbf{\Theta}_i\mathbf{U}^*\mathbf{\Theta}_i\big). \tag{43}$$

The quantity λ_i defined in (41) can be computed as follows, where $\mathbf{W}_i := \mathbf{\Theta}_i(\widehat{\mathbf{\Sigma}} + \rho\mathbf{U}^*)$:

$$\lambda_i := \left(p - 2\cdot\mathrm{tr}\left(\mathbf{W}_i\right) + \mathrm{tr}\left(\mathbf{W}_i^2\right)\right)^{1/2}. \tag{44}$$

Algorithm 3 summarizes the description above. Overall, this proximal-Newton (PN) algorithm *does not require any matrix inversions or Cholesky decompositions*. It only needs matrix-vector and matrix-matrix calculations, which might be attractive for different computational platforms (such as GPUs or simple parallel implementations). Note however that as we work through the dual problem, the primal solution can be dense even if majority of the entries are rather small (e.g., smaller than 10^{-6}).[5]

We now explain the underlying costs of each step in Algorithm 3, which is useful when we consider different strategies for the selection of the step size α_k. The computation of $\widetilde{\mathbf{Q}}$ and $\mathbf{\Delta}_i$ require basic matrix multiplications. For the computation of λ_i, we require two trace operations: $\mathrm{tr}(\mathbf{W}_i)$ in $\mathcal{O}(p)$ time-complexity and $\mathrm{tr}(\mathbf{W}_i^2)$ in $\mathcal{O}(p^2)$ complexity. We note here

5. In our MATLAB implementation below, we have not exploited the fact that the primal solutions are sparse. The overall efficiency can be improved via thresholding tricks, both in terms of time-complexity (e.g., less number of iterations) and matrix estimation quality.

Algorithm 3 (*Dual PN for graph selection* (DPNGS))

Input: Matrix $\widehat{\mathbf{\Sigma}} \succ 0$ and a given tolerance $\varepsilon > 0$. Set $\sigma := 0.25(5 - \sqrt{17})$.
Initialization: Find a starting point $\mathbf{\Theta}_0 \succ 0$.
for $i = 0$ **to** $i_{\max}$ **do**
1. Set $\widetilde{\mathbf{Q}} := \rho^{-1}\left(\mathbf{\Theta}_i\widehat{\mathbf{\Sigma}}\mathbf{\Theta}_i - 2\mathbf{\Theta}_i\right)$.
2. Compute $\mathbf{U}^*$ in (42).
3. Compute λ_i by (44), where $\mathbf{W}_i := \mathbf{\Theta}_i(\widehat{\mathbf{\Sigma}} + \rho\mathbf{U}^*)$.
4. If $\lambda_i \le \varepsilon$ terminate.
5. Compute $\mathbf{\Delta}_i := -\left((\mathbf{\Theta}_i\widehat{\mathbf{\Sigma}} - \mathbb{I})\mathbf{\Theta}_i + \rho\mathbf{\Theta}_i\mathbf{U}^*\mathbf{\Theta}_i\right)$.
6. If $\lambda_i > \sigma$, then set $\alpha_i := (1 + \lambda_i)^{-1}$. Otherwise, set $\alpha_i = 1$.
7. Update $\mathbf{\Theta}_{i+1} := \mathbf{\Theta}_i + \alpha_i\mathbf{\Delta}_i$.

end for

that, while $\mathbf{W}_i$ is a *dense* matrix, the trace operation in the latter case requires only the computation of the diagonal elements of $\mathbf{W}_i^2$. Given $\mathbf{\Theta}_i$, α_i and $\mathbf{\Delta}_i$, the calculation of $\mathbf{\Theta}_{i+1}$ has $\mathcal{O}(p^2)$ complexity. In contrast, evaluation of the objective can be achieved through Cholesky decompositions, which has $\mathcal{O}(p^3)$ time complexity.

To compute (42), we can use the fast proximal-gradient method (FPGM) (Nesterov, 2007; Beck and Teboulle, 2009a) with step size $1/L$ where L is the Lipschitz constant of the gradient of the objective function in (42). It is easy to observe that $L := \gamma^2_{\max}(\mathbf{\Theta}_i)$ where $\gamma_{\max}(\mathbf{\Theta}_i)$ is the largest eigenvalue of $\mathbf{\Theta}_i$. For sparse $\mathbf{\Theta}_i$, we can approximately compute $\gamma_{\max}(\mathbf{\Theta}_i)$ is $O(p^2)$ by using *iterative power methods* (typically, 10 iterations suffice). The projection onto $\|\mathbf{vec}(\mathbf{U})\|_\infty \le 1$ clips the elements by unity in $O(p^2)$ time. Since FPGM requires a constant number of iterations $k_{\max}$ (independent of p) to achieve an ε_{in} solution accuracy, the time-complexity for the solution in (42) is $O(k_{\max}M)$, where M is the cost of matrix multiplication. We have also implemented block coordinate descent and active set methods which scale $O(p^2)$ in practice when the solution is quite sparse.

Overall, the major operation with general proximal maps in the algorithm is typically the matrix-matrix multiplications of the form $\mathbf{\Theta}_i\mathbf{U}\mathbf{\Theta}_\mathbf{i}$, where $\mathbf{\Theta}_i$ and $\mathbf{U}$ are symmetric positive definite. This operation can naturally be computed (e.g., in a GPU) in a parallel or distributed manner. For more details of such computations we refer the reader to (Bertsekas and Tsitsiklis, 1989). It is important to note that without Cholesky decompositions used in objective evaluations, the basic DPNGS approach theoretically scales with the cost of matrix-matrix multiplications.

4.1.2 Proximal-Gradient Algorithm

Since $g(\mathbf{\Theta}) := \rho\,\|\mathbf{vec}(\mathbf{\Theta})\|_1$ and $\nabla f(\mathbf{\Theta}_i) = \mathbf{vec}(\widehat{\mathbf{\Sigma}} - \mathbf{\Theta}_i^{-1})$, the subproblem (15) becomes:

$$\mathbf{\Delta}_{i+1} := \mathcal{T}_{\tau_i\rho}\big(\mathbf{\Theta}_i - \tau_i(\widehat{\mathbf{\Sigma}} - \mathbf{\Theta}_i^{-1})\big) - \mathbf{\Theta}_i, \tag{45}$$

where $\mathcal{T}_\tau : \mathbb{R}^{p\times p} \to \mathbb{R}^{p\times p}$ is the component-wise matrix thresholding operator which is defined as $\mathcal{T}_\tau(\mathbf{\Theta}) := \max\{0, |\mathbf{\Theta}| - \tau\}$. We also note that the computation of $\mathbf{\Delta}_{i+1}$ requires a matrix inversion $\mathbf{\Theta}_i^{-1}$. Since $\mathbf{\Theta}_i$ is positive definite, one can apply Cholesky decompositions to compute $\mathbf{\Theta}_i^{-1}$ in $O(p^3)$ operations. To compute the quantity λ_i, we have $\lambda_i := \|\mathbf{\Delta}_i\|_{\mathbf{\Theta}_i} =$

$\left\|\boldsymbol{\Theta}_i^{-1}\boldsymbol{\Delta}_i\right\|_2$. We also choose $L_i := 0.5\|\nabla^2 f(\boldsymbol{\Theta}_i)\|_2 = 0.5\|\boldsymbol{\Theta}_i^{-1}\|_2^2$. The above are summarized in Algorithm 4.

Algorithm 4 (*Proximal-gradient method for graph selection* (`ProxGrad1`))

Initialization: Choose a starting point $\boldsymbol{\Theta}_0 \succ 0$.
for $i = 0$ **to** $i_{\max}$ **do**
1. Compute $\boldsymbol{\Theta}_i^{-1}$ via Cholesky decomposition.
2. Choose L_i satisfying (30) and set $\tau_i := L_i^{-1}$.
3. Compute the search direction $\boldsymbol{\Delta}_i$ as (45).
4. Compute $\beta_i := L_i\|\mathbf{vec}(\boldsymbol{\Delta}_i)\|_2$ and $\lambda_i := \|\boldsymbol{\Theta}_i^{-1}\boldsymbol{\Delta}_i\|_2$.
5. Determine the step size $\alpha_i := \frac{\beta_i}{\lambda_i(\lambda_i+\beta_i)}$.
6. Update $\boldsymbol{\Theta}_{i+1} := \boldsymbol{\Theta}_i + \alpha_i\boldsymbol{\Delta}_i$.

end for

The per iteration complexity is dominated by matrix-matrix multiplications and Cholesky decompositions for matrix inversion calculations. In particular, Step 1 requires a Cholesky decomposition with $\mathcal{O}(p^3)$ time-complexity. Step 2 requires to compute ℓ_2-norm of a symmetric positive matrix, which can be done by a power-method in $\mathcal{O}(p^2)$ time-complexity. The complexity of Steps 3, 4 and 6 requires $\mathcal{O}(p^2)$ operations. Step 2 may require additional bisection steps as mentioned in Algorithm 2 whenever $\lambda_k < 1$.

4.2 Poisson Intensity Reconstruction

We now describe a variant of Algorithm 2; a similar instance based on Algorithm 1 can be easily devised and we omit the details here. First, we can easily check that the function $\tilde{f}(\mathbf{x}) := \sum_{i=1}^m \left(\mathbf{a}_i^T\mathbf{x} - y_i\log(\mathbf{a}_i^T\mathbf{x})\right)$ in (3) is convex and self-concordant with parameter $M_{\tilde{f}} := 2\cdot\max\left\{\frac{1}{\sqrt{y_i}} \mid y_i > 0, i = 1,\ldots,m\right\}$, see (Nesterov, 2004, Theorem 4.1.1). We define the functions f and g as

$$f(\mathbf{x}) := \frac{M_{\tilde{f}}^2}{4}\tilde{f}(\mathbf{x}), \quad g(\mathbf{x}) := \frac{M_{\tilde{f}}^2}{4}\left(\rho\phi(\mathbf{x}) + \delta_{\{\mathbf{u} \mid \mathbf{u}\geq 0\}}(\mathbf{x})\right), \tag{46}$$

where f and g satisfy Assumption 1 and $\delta_{\mathcal{C}}$ is the indicator function of $\mathcal{C}$. Thus, the problem in (3) can be equivalently transformed into (1). Here, the gradient and the Hessian of f satisfy:

$$\nabla f(\mathbf{x}) = \frac{M_{\tilde{f}}^2}{4}\sum_{i=1}^m\left(1 - \frac{y_i}{\mathbf{a}_i^T\mathbf{x}}\right)\mathbf{a}_i \quad\text{and}\quad \nabla^2 f(\mathbf{x}) = \frac{M_{\tilde{f}}^2}{4}\sum_{i=1}^m\frac{y_i}{(\mathbf{a}_i^T\mathbf{x})^2}\mathbf{a}_i\mathbf{a}_i^T, \tag{47}$$

respectively. For a given vector $\mathbf{d}\in\mathbb{R}^n$, the local norm $\|\mathbf{d}\|_{\mathbf{x}}$ can then be written as

$$\|\mathbf{d}\|_{\mathbf{x}} := \left(\mathbf{d}^T\nabla^2 f(\mathbf{x})\mathbf{d}\right)^{1/2} = \frac{M_{\tilde{f}}}{2}\left(\sum_{i=1}^m\frac{y_i(\mathbf{a}_i^T\mathbf{d})^2}{(\mathbf{a}_i^T\mathbf{x})^2}\right)^{1/2}. \tag{48}$$

Computing this quantity requires one matrix-vector multiplication and $\mathcal{O}(m)$ operations.

For the Poisson model, the subproblem (15) is expressed as follows:

$$\min_{\mathbf{x}\geq 0}\left\{(1/2)\|\mathbf{x}-\mathbf{w}^k\|_2^2+\rho_k\phi(\mathbf{x})\right\}, \tag{49}$$

where $\mathbf{w}^k := \mathbf{x}^k - L_k^{-1}\nabla f(\mathbf{x}^k)$ and $\rho_k := \frac{\rho M_{\tilde{f}}^2}{4L_k}$. As a penalty function ϕ in the Poisson intensity reconstruction, we use the Total Variation norm (TV-norm), defined as $\phi(\mathbf{x}) := \|\mathbf{D}\mathbf{x}\|_1$ (isotropic) or $\phi(\mathbf{x}) := \|\mathbf{D}\mathbf{x}\|_{1,2}$ (anti-isotropic), where $\mathbf{D}$ is a forward linear operator (Chambolle and Pock, 2011; Beck and Teboulle, 2009b). For both TV-norm regularizers, the method proposed in (Beck and Teboulle, 2009b) can solve (49) efficiently.

The above discussion leads to Algorithm 5. We note that the constant L_k at Step 2 of this algorithm can be estimated based on different rules. In our implementation below, we initialize L_k at a Barzilai-Borwein step size, i.e., $L_k := \frac{(\nabla f(\mathbf{x}^k)-\nabla f(\mathbf{x}^{k-1}))^T(\mathbf{x}^k-\mathbf{x}^{k-1})}{\|\mathbf{x}^k-\mathbf{x}^{k-1}\|_2^2}$ and may perform a few backtracking iterations on L_k to ensure the condition (30) whenever $\lambda_k < 1$.

Algorithm 5 (`ProxGrad` *for Poisson intensity reconstruction* (`ProxGrad2`))

Inputs: $\mathbf{x}^0 \geq 0$, $\varepsilon > 0$ and $\rho > 0$.
Compute $M_{\tilde{f}} := 2\max\left\{\frac{1}{\sqrt{\mathbf{y}_i}} \mid \mathbf{y}_i > 0, i = 1,\ldots,m\right\}$.
for $k = 0$ **to** $k_{\max}$ **do**
1. Evaluate the gradient of f as (47).
2. Compute an appropriate value $L_k > 0$ that satisfies (30).
3. Compute $\rho_k := 0.25\rho M_{\tilde{f}}^2 L_k^{-1}$ and $\mathbf{w}^k := \mathbf{x}^k - L_k^{-1}\nabla f(\mathbf{x}^k)$.
4. Compute $\mathbf{s}_g^k$ by solving (49) and then compute $\mathbf{d}_g^k := \mathbf{s}_g^k - \mathbf{x}^k$.
5. Compute $\beta_k := L_k\|\mathbf{d}_g^k\|_2^2$ and $\lambda_k := \|\mathbf{d}_g^k\|_{\mathbf{x}^k}$ as (48).
6. If $e_k := L_k^{-1}\sqrt{\beta_k} \leq \varepsilon$ then terminate.
7. Determine the step size $\alpha_k := \frac{\beta_k}{\lambda_k(\lambda_k+\beta_k)}$.
8. Update $\mathbf{x}^{k+1} := \mathbf{x}^k + \alpha_k\mathbf{d}_g^k$.

end for

Note that we can modify Step 8 in Algorithm 5 by using the update scheme (36) to obtain a new variant of this algorithm. We omit the details here.

4.3 Heteroscedastic LASSO

We focus on a convex formulation of the unconstrained LASSO problem with unknown variance studied in (Städler et al., 2012) as

$$(\boldsymbol{\beta}^*,\sigma^*) := \underset{\boldsymbol{\beta}\in\mathbb{R}^p,\sigma\in\mathbb{R}_{++}}{\arg\min}\left\{-\log(\sigma)+(1/(2n))\,\|\mathbf{X}\boldsymbol{\beta}-\sigma\mathbf{y}\|_2^2+\rho\,\|\boldsymbol{\beta}\|_1\right\}. \tag{50}$$

However, our algorithm can be applied to solve the multiple unknown variance case considered in (Dalalyan et al., 2013).

By letting $\mathbf{x} := (\boldsymbol{\beta}^T,\sigma)^T \in \mathbb{R}^{p+1}$, $f(\mathbf{x}) := -\log(\sigma)+(1/(2n))\,\|\mathbf{X}\boldsymbol{\beta}-\sigma\mathbf{y}\|_2^2$. Then, it is easy to see that the function f is standard self-concordant. Hence, we can apply Algorithm 2 to solve this problem. To highlight the salient differences in the code, we note the following:

- Define $\mathbf{z} := \mathbf{X}\boldsymbol{\beta} - \sigma\mathbf{y}$, then the gradient vector of function f can be computed as

$$\nabla f(\mathbf{x}) := \left(n^{-1}\mathbf{z}^T\mathbf{X}, -\sigma^{-1} - n^{-1}\mathbf{y}^T\mathbf{z}\right)^T.$$

  This computation requires two matrix-vector multiplications and one inner product.

- The quantity λ_k can be explicitly computed as

$$\lambda_k := \left(\left(\sigma_k^{-2} + n^{-1}\mathbf{y}^T\mathbf{y}\right)(\mathbf{d}_\sigma^k)^2 + n^{-1}\mathbf{z}_k^T\mathbf{z}_k - 2n^{-1}\mathbf{d}_\sigma^k\mathbf{y}^T\mathbf{z}_k\right)^{1/2},$$

  where $\mathbf{z}_k := \mathbf{X}\mathbf{d}_{\boldsymbol{\beta}}^k$ and $\mathbf{d}_g^k := ((\mathbf{d}_{\boldsymbol{\beta}}^k)^T, \mathbf{d}_\sigma^k)^T$ is the search direction. This quantity requires one matrix-vector multiplication and two inner products. Moreover, this matrix-vector product can be reused to compute the gradient for the next iteration.

The final algorithm is very similar to Algorithm 5 and hence we omit the details.

5. Numerical Experiments

In this section, we illustrate our optimization framework via numerical experiments on the variants discussed in Section 4. We only focus on proximal gradient and Newton variants and encourage the interested reader to try out the quasi-Newton variants for their own applications. All the tests are performed in MATLAB 2011b running on a PC Intel Xeon X5690 at 3.47GHz per core with 94Gb RAM.[6]

5.1 Proximal-Newton Method in Action

By using the graph selection problem, we first show that the modifications on the proximal-Newton method provides advantages in practical convergence as compared to state-of-the-art strategies and provides a safeguard for line-search procedures in optimization routines. We then highlight the impact of different subsolvers for (37) in the practical convergence of the algorithms.

5.1.1 Comparison of Different Step-Size Selection Procedures

We apply four different step-size selection procedures in our proximal-Newton framework to solve problem (2). Specifically, we test the algorithm based on the following configuration:

(*i*) We implement Algorithm 3 in MATLAB using FISTA (Beck and Teboulle, 2009a) to solve the dual subproblem with the following stopping criterion: $\|\boldsymbol{\Theta}_{i+1} - \boldsymbol{\Theta}_i\|_F \leq 10^{-8} \times \max\{\|\boldsymbol{\Theta}_{i+1}\|_F, 1\}$.

(*ii*) We consider four different globalization procedures, whose details can be found in Section 3.1: *a*) `NoLS` which uses the analytic step size $\alpha_k^* = (1+\lambda_k)^{-1}$, *b*) `BtkLS` which is an instance of the proximal-Newton framework of (Lee et al., 2012) and uses the standard backtracking line-search based on Armijo's rule, *c*) `E-BtkLS` which is based on the standard backtracking line-search enhanced by the lower bound α_k^* and, *d*)

6. We also provide MATLAB implementations of the examples in this section as a software package (SCOPT) at http://lions.epfl.ch/software.

`FwLS` as the forward line-search by starting from α_k^* and increasing the step size until either $\alpha_k = 1$, infeasibility or the objective value does not improve.

(*iii*) We test our implementation on four problem cases: The first problem is a synthetic examples of size $p = 10$, where the data is generated as in (Kyrillidis and Cevher, 2013). We run this test for 10 times and report computational primitives in average. Three remaining problems are based on real data from `http://ima.umn.edu/~maxxa007/send_SICS/`, where the regularization parameters are chosen as the standard values (cf., Tran-Dinh et al. (2013a); Lee et al. (2012); Hsieh et al. (2011)). We terminate the proximal-Newton scheme if $\lambda_k \leq 10^{-6}$.

The numerical results are summarized in Table 2. Here, #iter denotes the (average) number of iterations, #chol represents the (average) number of Cholesky decompositions and #Mm is the (average) number of matrix-matrix multiplications.

	Synthetic ($\rho = 0.01$)			Arabidopsis ($\rho = 0.5$)			Leukemia ($\rho = 0.1$)			Hereditary ($\rho = 0.1$)		
LS Scheme	#iter	#chol	#Mm	#iter	#chol	#Mm	#iter	#chol	#Mm	#iter	#chol	#Mm
`NoLS`	25.4	-	3400	18	-	1810	44	-	9842	72	-	20960
`BtkLS`	25.5	37.0	2436	11	25	718	15	50	1282	19	63	2006
`E-BtkLS`	25.5	36.2	2436	11	24	718	15	49	1282	15	51	1282
`FwLS`	18.1	26.2	1632	10	17	612	12	34	844	14	44	1126

Table 2: Metadata for the line search strategy comparison

We can see that our new step-size selection procedure `FwLS` shows superior empirical performance as compared to the rest: The standard approach `NoLS` usually starts with pessimistic step-sizes which are designed for worst-case problem structures. Therefore, we find it advantageous to continue with a forward line-search procedure. Whenever it reaches the quadratic convergence, no Cholesky decompositions are required. This makes a difference, compared to standard backtracking line-search `BtkLS` where we need to evaluate the objective value at every iteration. While there is no free lunch, the cost of computing λ_k is $\mathcal{O}(p^2)$ in `FwLS`, which turns out to be quite cheap in this application. The `E-BtkLS` combines both backtrack line-search and our analytic step-size $\alpha_k^* := (1 + \lambda_k)^{-1}$, which outperforms `BtkLS` as the regularization parameter becomes smaller. Finally, we note that the `NoLS` variant needs more iterations but it does not require any Cholesky decompositions, which might be advantageous in homogeneous computational platforms.

5.1.2 Impact of Different Solvers for the Subproblems

As mentioned in the introduction, an important step in our second order algorithmic framework is the solution of the subproblem (15). If the variable matrix $\mathbf{H}_k$ is not diagonal, computing $\mathbf{s}_{\mathbf{H}_k}^k$ corresponds to solving a convex subproblem. For a given regularization term g, we can exploit different existing approaches to tackle this problem. We illustrate that the overall framework to be quite robust against the solution accuracy of the individual subsolver.

In this test, we consider the broad used ℓ_1-norm function as the regularizer. Hence, (15) collapses to an unconstrained LASSO problem; cf. (Wright et al., 2009). To this end, we implement the proximal-Newton algorithm to solve the graph learning problem (2) where

Sub-solvers	Estrogen ($p = 692$)			Arabidopsis ($p = 834$)			Leukemia ($p = 1255$)			Hereditary($p = 1869$)		
	#iter	#chol	time[s]	#iter	#chol	time[s]	#iter	#chol	time[s]	#iter	#chol	time[s]
	$\rho = 0.5$											
	#nnz $= 0.022p^2$			#nnz $= 0.030p^2$			#nnz $= 0.022p^2$			#nnz $= 0.020p^2$		
pFISTA	9	29	13.10	10	35	24.76	9	31	286.57	17	80	1608.66
pFISTA[gpu]	9	29	10.70	10	35	16.81	9	31	231.97	17	80	1265.97
dFISTA	8	16	4.66	10	17	10.92	14	22	50.19	14	27	147.86
dFISTA[gpu]	8	16	4.16	10	17	7.89	14	22	43.53	14	27	120.16
FastAS	7	24	28.69	8	27	96.93	9	31	532.11	11	40	1682.28
BCDC	8	25	90.35	9	28	227.27	9	31	549.80	12	47	3452.82
MatQUIC	11	29	21.61	10	35	50.67	10	35	119.06	14	44	891.29
ProxGrad1	175	175	8.82	226	226	17.78	230	230	44.06	660	660	350.52
	$\rho = 0.1$											
	#nnz $= 0.072p^2$ ($\sim 6\%$)			#nnz $= 0.074p^2$			#nnz $= 0.065p^2$			#nnz $= 0.063p^2$		
pFISTA	34	101	357.25	57	148	1056.90	143	242	7490.27	-	-	-
pFISTA[gpu]	34	101	300.90	57	148	730.07	143	242	6083.06	-	-	-
dFISTA	14	32	12.51	12	35	15.53	12	34	38.73	14	44	150.03
dFISTA[gpu]	14	32	11.18	12	35	11.18	12	34	33.45	14	44	121.37
FastAS	-	-	-	-	-	-	-	-	-	-	-	-
BCDC	13	48	1839.17	15	50	4806.62	-	-	-	-	-	-
MatQUIC	30	88	573.87	36	95	1255.13	36	95	4260.97	-	-	-
ProxGrad1	4345	4345	224.95	6640	6640	532.77	9225	9225	1797.49	-	-	-

Table 3: Metadata for the subsolver efficiency comparison

$g(\mathbf{x}) := \rho \|\mathbf{x}\|_1$. To show the impact of the subsolver in (2), we implement the following methods, which are all available in our software package SCOPT:

(i) `pFISTA` and `dFISTA`: in these cases, we use the FISTA algorithm (Beck and Teboulle, 2009a) for solving the primal (37) and the dual subproblem (39). Moreover, to speedup the computations, we further run these methods on the GPU [NVIDIA Quadro 4000].

(ii) `FastAS`: this method corresponds to the exact implementation of the fast active-set method proposed in (Kim and Park, 2010) for solving the primal-dual (37).

(iii) `BCDC`: here, we consider the block-coordinate descent method implemented in (Hsieh et al., 2011) for solving the primal subproblem (37).

We also compare the above variants of the *proximal-Newton approach* with (i) the proximal-gradient method (Algorithm 4) denoted by `ProxGrad1` and (ii) a precise MATLAB implementation of QUIC (`MatQUIC`), as described in (Hsieh et al., 2011). For the proximal-Newton and `MatQUIC` approaches, we terminate the execution if the maximum number of iterations exceeds 200 or the total execution time exceeds the 5 hours. The maximum number of iterations in `ProxGrad1` is set to 10^4.

The results are reported in Table 3. Overall, we observe that `dFISTA` shows superior performance across the board in terms of computational time and the total number of Cholesky decompositions required. Here, #nnz represents the number of nonzero entries in the final solution. The notation "$-$" indicates that the algorithms exceed either the maximum number of iterations or the time limit (5 hours).

If the parameter ρ is relatively large (i.e., the solution is expected to be quite sparse), `FastAS`, `BCDC` and `MatQUIC` perform well and converge in a reasonable time. This is expected since all three approaches vastly rely on the sparsity of the solution: the sparser the solution is, the faster their computations are performed, as restricted on the active set of variables. However, when ρ is small, the performance of these methods significantly degrade due to the increased number of active (non-zero) entries.

Aside from the above, `ProxGrad1` performs well in terms of computational time, as compared to the rest of the methods. Unfortunately, the number of Cholesky decompositions in this method can become as many as the number of iterations, which indicates a computational bottleneck in high-dimensional problem cases. Moreover, when ρ is small, this method also slows down and requires more iterations to converge.

On the other hand, we also note that `pFISTA` is rather sensitive to the accuracy of the subsolver within the quadratic convergence region. In fact, while `pFISTA` reaches medium scale accuracies in a manner similar to `dFISTA`, it spends most of its iterations trying to achieve the higher accuracy values.

5.2 Proximal-Gradient Algorithm in Action

In this subsection, we illustrate the performance of proximal gradient algorithm in practice on various problems with different regularizers.

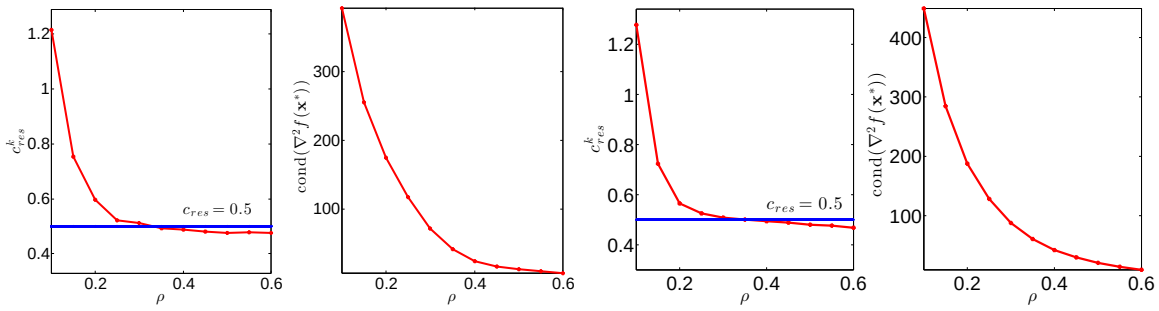
5.2.1 Linear Convergence

To show the linear convergence of `ProxGrad1` (Algorithm 2) in practice, we consider the following numerical test. Our experiment is based on the `Lymph` and `Estrogen` problems downloaded from `http://ima.umn.edu/~maxxa007/send_SICS/`. For both problem cases, we use different values for ρ as $\rho = [0.1 : 0.05 : 0.6]$ in MATLAB notation. For each configuration, we measure the quantity:

$$c^k_{\rm res} := \frac{\big\|(\mathbf{D}_k - \nabla^2 f(\mathbf{x}^*))\mathbf{d}^k_g\big\|^*_{\mathbf{x}^*}}{\|\mathbf{d}^k_g\|_{\mathbf{x}^*}}, \tag{51}$$

for few last iterations. This quantity can be referred to as the restricted approximation gap of D_k to $\nabla^2 f(\mathbf{x}^*)$ along the proximal-gradient direction $\mathbf{d}^k_g$. We first run the proximal-Newton method up to 10^{-16} accuracy to obtain the solution $\mathbf{x}^*$ and then run the proximal-gradient algorithm up to 10^{-8} accuracy to compute $c^k_{\rm res}$ and the norm $\|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*}$. From the proof of Theorem 14, we can show that if $\mathrm{c}^k_{\rm res} < 0.5$ for sufficiently large k, then the sequence $\{\mathbf{x}^k\}_{k\geq 0}$ locally converges to $\mathbf{x}^*$ at a linear rate. We note that this condition is much weaker than the last condition given in Theorem 14 but more difficult to interpret. Note that the requirement in Theorem 14 leads to a restriction on the condition number of $\nabla^2 f(\mathbf{x}^*)$ to be less than 3. We perform this test on two problem instances with 11 different values of the regularization parameter and then compute the median of $c^k_{\rm res}$ for each problem. Figure 4 shows the median of the restricted approximation gap $c^k_{\rm res}$ and the real condition number of $\nabla^2 f(\mathbf{x}^*)$, respectively.

As expected, we observe that the real condition number of $\nabla^2 f(\mathbf{x}^*)$ increases as the regularization parameter decreases. Moreover, the last condition given in Theorem 14 does not hold in this example. However, if we look at the restricted condition number computed



(a) `Lymph` data set ($p = 578$) (b) `Estrogen` data set ($p = 692$)

Figure 4: For each test case: **(Left)** Restricted approximation gap c_{res}^k **(Right)** The actual condition number of $\nabla^2 f(\mathbf{x}^*)$.

by (51), we can observe that for $\rho \gtrsim 0.3$, this value is strictly smaller than 0.5. In this case, the local linear convergence is actually observed in practice.

While $c_{\text{res}}^k < 0.5$ is only a sufficient condition and can possibly be improved, we find it to be a good indicator of the convergence behavior. Figure 5 shows the last 100 iterations of our gradient method for the `Lymph` problem with $\rho = 0.15$ and $\rho = 0.55$. The number of iterations needed to achieve the final solution in these cases is 1525 and 140, respectively. In the former case, the calculated restricted condition number is above 0.5 and the final convergence rate suffers. For instance, the contraction factor κ in the estimate $\left\|\mathbf{x}^{k+1} - \mathbf{x}^*\right\|_{\mathbf{x}^*} \leq \kappa \left\|\mathbf{x}^k - \mathbf{x}^*\right\|_{\mathbf{x}^*}$ is close to 1 when $\rho = 0.15$, while it is smaller when $\rho = 0.55$. We can observe from Figure 5 (left) that the error $\|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*}$ drops rapidly at the last few iterations due to the affect of the bisection procedure, where we check the condition (30) for $\lambda_k < 1$.

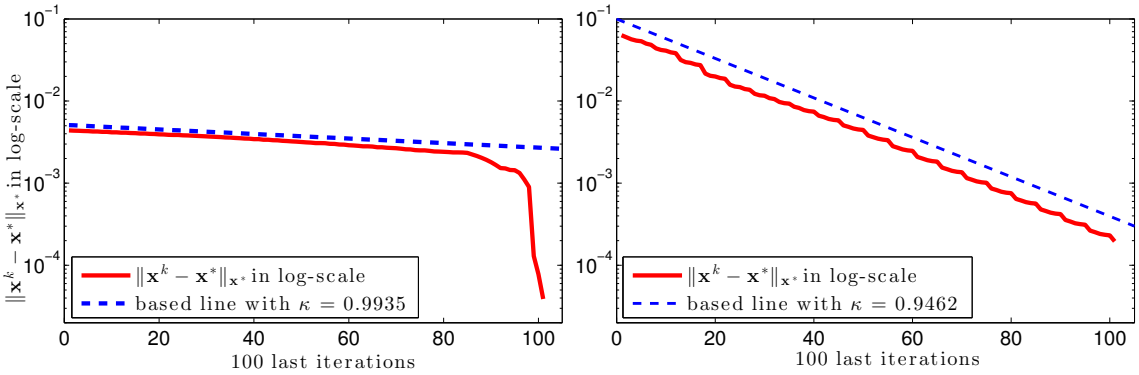


Figure 5: Linear convergence of `ProxGrad1` for `Lymph`: **Left**: $\rho = 0.15$ and **Right**: $\rho = 0.55$.

5.2.2 TV_{ℓ_1}-regularizer

In this experiment, we consider the Poisson intensity reconstruction problem, where the regularizer g, the TV_{ℓ_1}-norm which is called the *anisotropic*-TV; as an example, cf. (Beck and Teboulle, 2009b). Hence, we implement Algorithm 5 (`ProxGrad2`) to solve (3), improve it using the greedy step-size modification as described in Section 3.3 (`ProxGrad2g`), and

compare its performance with the state-of-the-art Sparse Poisson Intensity Reconstruction Algorithms (SPIRAL-TAP) toolbox (Harmany et al., 2012).

As a termination criterion, we have $\|\mathbf{d}_g^k\|_2 \leq 10^{-5}\max\{1, \|\mathbf{x}^k\|_2\}$ or when the objective value does not significantly change after 5 successive iterations, i.e., for each k, $\left|f(\mathbf{x}^{k+j}) - f(\mathbf{x}^k)\right| \leq 10^{-8}\max\{1, \left|f(\mathbf{x}^k)\right|\}$ for $j = 1, \ldots, 5$.

We first illustrate the convergence behavior of the three algorithms under comparison. We consider two image test cases: `house` and `cameraman`, and we set the regularization parameter of the TV_{ℓ_1}-norm to $\rho = 2.5 \times 10^{-5}$. Figure 9 illustrate the convergence of the algorithms both in iteration count and the timing.

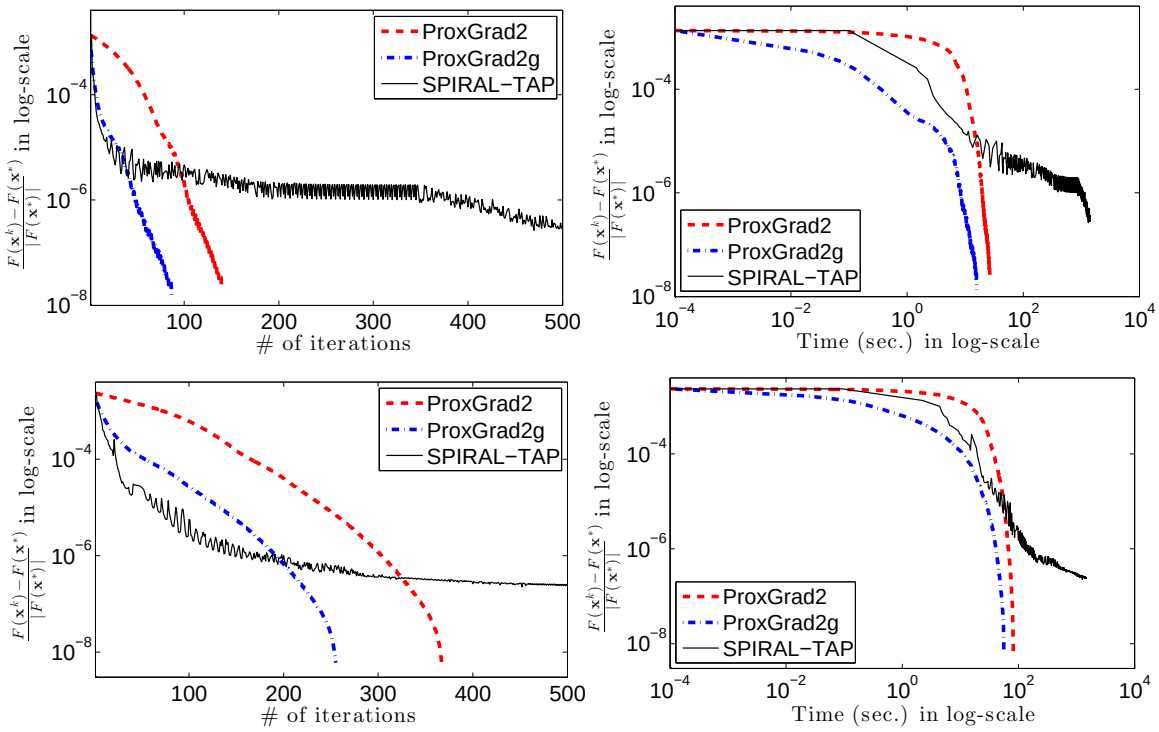


Figure 6: Convergence of three algorithms for `house` (top) and `cameraman` (bottom). **Left**: in iteration scale **Right**: in time log-scale.

Overall, `ProxGrad2g` exhibits the best convergence behavior in terms of iterations and time. Due to the inaccurate solutions of the subproblem (49), the methods might exhibit oscillations. Since SPIRAL-TAP employs a Barzilai-Borwein step-size and performs a line-search procedure up to very small step-size, the objective value is not sufficiently decreased; as a result of this, we observe more oscillations in the objective value.

In stark contrast, `ProxGrad2` and `ProxGrad2g` use the Barzilai-Borwein step-size as an initial-guess for computing a search direction and then use the step-size correction procedure to ensure that the objective function decreases a certain amount at each iteration. This strategy turns out to be more effective since milder oscillations in the objective values are observed in practice (which are due to the inaccuracy of the TV-proximal operator).

Finally, we test the performance of `ProxGrad2`, `ProxGrad2g` and `SPIRAL-TAP` on 4 different image cases: `barbara, cameraman, house` and `lena`. We set ρ to two different values: $\rho \in \{10^{-5},\ 2.5 \cdot 10^{-5}\}$. These values are chosen in order to obtain the best visual

Figure 7: The reconstructed images for `barbara` ($\rho = 2.5 \times 10^{-5}$)

reconstructions (e.g., see Figure 7) and are previously used in (Harmany et al., 2012). The summary results reported in Table 4. Here, AC denotes the multiplicative factor in time acceleration of `ProxGrad2` as compared to `SPIRAL-TAP`, and ΔF is the difference between the corresponding obtained objective values between `ProxGrad2` and `SPIRAL-TAP` (a positive ΔF means that `SPIRAL-TAP` obtains a higher objective value at termination).

	`ProxGrad2g / ProxGrad2 / SPIRAL-TAP`											
Image	$\rho \times 10^{-5}$	#iteration			CPU time [s]			AC		$F^k_{\min}$	ΔF	
`house`	1.0	116	256	500	27.45	56.95	1658.00	60	29	-10718352.93	0.31	0.70
(256 × 256)	2.5	92	244	500	18.18	50.26	1431.94	79	28	-10711758.80	3.20	3.32
`barbara`	1.0	200	324	500	46.92	77.77	1204.36	26	15	-7388497.47	0.05	0.30
(256 × 256)	2.5	164	268	500	36.45	67.98	1620.95	44	24	-7377424.50	1.90	2.02
`cameraman`	1.0	396	516	500	99.56	117.75	389.79	4	3	-9186631.65	0.19	0.07
(256 × 256)	2.5	256	368	500	59.75	85.25	1460.62	24	17	-9175307.33	2.29	2.31
`lena`	1.0	152	220	500	27.43	41.31	1212.69	44	29	-5797053.79	0.10	0.10
(204 × 204)	2.5	304	184	500	59.20	36.77	1132.04	19	31	-5789554.53	1.52	1.25

Table 4: The results and performance of three algorithms

From Table 4 we observe that `ProxGrad2` and `ProxGrad2g` are superior to `SPIRAL-TAP`, both in terms of CPU time and the final objective value in majority of problems. As the table shows, `ProxGrad2g` can be 4 to 79 times faster than `SPIRAL-TAP`. Moreover, it reports a better objective values in all cases.

5.2.3 A Comparison to Standard Gradient Methods Based on $\mathcal{F}_L$ Assumption

In this subsection, we use the LASSO problem (50) with unknown variance as a simple test case to illustrate the improvements over the "standard" methods. Note that the standard Lipschitz gradient assumption no longer holds in this example due to the log-term $\log(\sigma)$. For this comparison, we dub our algorithm as `ProxGrad3(g)` and compare it against a state-of-the-art TFOCS software package (Becker et al., 2011). The input data is synthetically generated based on the linear model $\mathbf{y} = \mathbf{X}\boldsymbol{\beta}^* + \mathbf{s}$, where $\boldsymbol{\beta}$ is the true sparse parameter vector; $\mathbf{X}$ is a Gaussian $n \times p$ matrix and $\mathbf{s} \sim \mathcal{N}(0, \sigma^2)$, where $\sigma = 0.01$. In TFOCS, we configure the Nesterov's accelerated algorithm with two proximal operations (TFOCS-N07) and adaptive restart as well as the standard gradient method (TFOCS-GRA). Both options

use a backtracking step-size selection procedure due to the presence of the logarithmic term in the objective.

As we can see in Figure 9 and Table 5 that `ProxGrad3g` performs the best and manages to converge to a high accuracy solution at a linear rate in both examples. Interestingly, we find the per iteration complexity of `ProxGrad3g` is similar to `ProxGrad3` and TFOCS-GRA. In terms of per iteration cost, TFOCS-N07 is the most expensive one as it uses dual prox operations and adaptive restart, and requires more backtracking operations. Hence, while it takes less iterations as compared to the TFOCS-GRA, it performs worse in terms of timing. For illustration purposes, we ran the algorithms to high accuracy. However, if a typical stopping criteria such as 10^{-6} is used, our algorithm `ProxGrad3g` obtains $\times 3$ to $\times 8$ speed-ups over the standard gradient algorithm with backtracking enhancements.

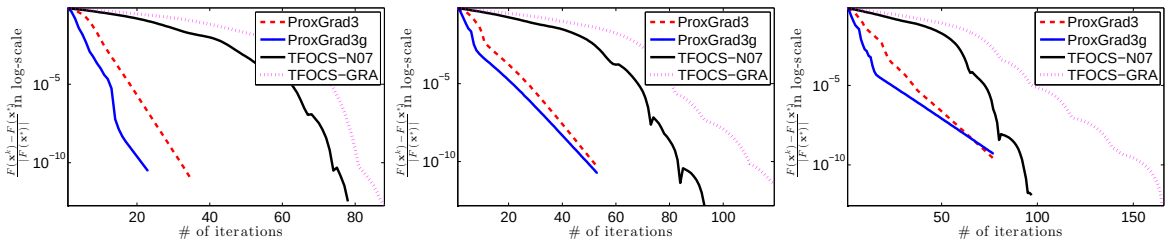


Figure 8: Convergence plots of algorithms under comparison for $n = 3000$ and $p = 10000$. From left to right, $\rho = 10^{-3}, \frac{2}{3} \cdot 10^{-4}, 5 \cdot 10^{-4}$.

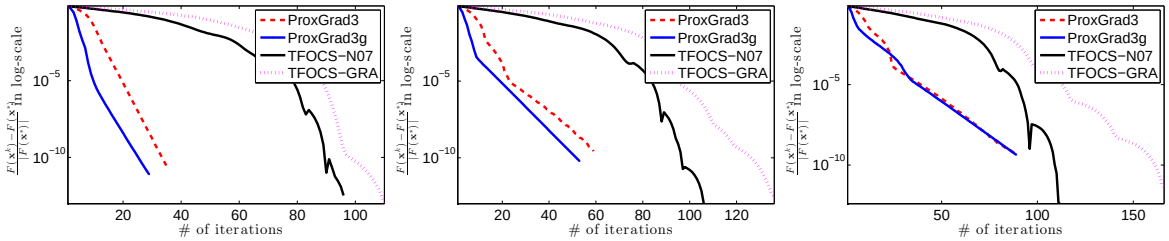


Figure 9: Convergence plots of algorithms under comparison for $n = 15000$ and $p = 50000$. From left to right, $\rho = 2 \cdot 10^{-4}, \frac{4}{3} \cdot 10^{-4}, 10^{-4}$.

6. Conclusions

We propose a variable metric method for minimizing convex functions that are compositions of proximity functions with self-concordant smooth functions. Our framework does not rely on the usual Lipschitz gradient assumption on the smooth part for its convergence theory. A highlight of this work is the new set of analytic step-size selection and correction procedures, which are best matched to the underlying problem structures. Our empirical results illustrate that the new theory leads to significant improvements in the practical performance of the algorithmic instances when tested on a variety of different applications.

In this work, we present a convergence proof for composite minimization problems under the assumption of *exact algorithmic calculations* at each step of the methods. As future research direction, an interesting problem to pursue is the extension of this analysis to

Problem	ProxGrad3 / ProxGrad3g / TFOCS-N07 / TFOCS-GRA										
$(3000, 10000)$	#iteration				CPU time [s]				$\|\boldsymbol{\beta}\|_0$	$\|\widehat{\boldsymbol{\beta}}\|_0$	Overlap (%)
$\rho = 10^{-3}$	36	24	79	88	1.0096	0.7862	3.2759	1.7648		166	44.72
$\rho = \frac{2}{3} \cdot 10^{-4}$	54	54	94	119	1.2974	1.2918	3.6499	2.4002	360	378	92.22
$\rho = 5 \cdot 10^{-4}$	78	78	97	166	1.7420	1.7513	3.7794	3.3416		412	100
$(15000, 50000)$	#iteration				CPU time [s]				$\|\boldsymbol{\beta}\|_0$	$\|\widehat{\boldsymbol{\beta}}\|_0$	Overlap (%)
$\rho = 2 \cdot 10^{-4}$	36	30	99	110	21.7937	19.3241	82.3298	46.0475		845	44.98
$\rho = \frac{4}{3} \cdot 10^{-4}$	60	54	108	136	31.7884	29.1194	89.4279	57.9088	1800	1886	87.91
$\rho = 10^{-4}$	90	90	113	166	44.2692	44.0611	95.3060	70.0946		2201	100

Table 5: Metadata on the Lasso problem with unknown variance

include *inexact calculations* and study how these errors propagate into the convergence and convergence rate guarantees (Kyrillidis et al., 2014). We hope this paper triggers future efforts along this direction.

Acknowledgments

This work is supported in part by the European Commission under Grant MIRG-268398, ERC Future Proof and SNF 200021-132548, SNF 200021-146750 and SNF CRSII2-147633. The authors are also grateful to three anonymous reviewers as well as to the action editor for their thorough reviews of this work, comments and suggestions on improving the content and the presentation of this paper.

Appendix A. Technical proofs

In this appendix, we provide the detailed proofs of the theoretical results in the main text. It consists of *global convergence* and *local convergence rate* of our algorithms and other technical proofs.

A.1 Proof of Lemma 4

Since g is convex, we have $g(\mathbf{y}) \geq g(\mathbf{x}) + \mathbf{v}^T(\mathbf{y} - \mathbf{x})$ for all $\mathbf{v} \in \partial g(\mathbf{x})$. By adding this inequality to (9) and noting that $F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x})$, we obtain

$$\begin{aligned} F(\mathbf{y}) &\geq F(\mathbf{x}) + (\nabla f(\mathbf{x}) + \mathbf{v})^T (\mathbf{y} - \mathbf{x}) + \omega(\|\mathbf{y} - \mathbf{x}\|_{\mathbf{x}}) \\ &\geq F(\mathbf{x}) - \lambda(\mathbf{x}) \|\mathbf{y} - \mathbf{x}\|_{\mathbf{x}} + \omega(\|\mathbf{y} - \mathbf{x}\|_{\mathbf{x}}). \end{aligned} \tag{52}$$

Here, the last inequality is due to the generalized Cauchy-Schwartz inequality and $\lambda(\mathbf{x}) := \|\nabla f(\mathbf{x}) + \mathbf{v}\|_{\mathbf{x}}^*$. Let $\mathcal{L}_F(F(\mathbf{x})) := \{\mathbf{y} \in \mathrm{dom}(F) \mid F(\mathbf{y}) \leq F(\mathbf{x})\}$ be a sublevel set of F. Then, for any $\mathbf{y} \in \mathcal{L}_F(F(\mathbf{x}))$, we have $F(\mathbf{y}) \leq F(\mathbf{x})$ which leads to

$$\lambda(\mathbf{x}) \|\mathbf{y} - \mathbf{x}\|_{\mathbf{x}} \geq \omega(\|\mathbf{y} - \mathbf{x}\|_{\mathbf{x}}),$$

due to (52). Let $s(t) := \frac{\omega(t)}{t} = 1 - \frac{\ln(1+t)}{t}$. The last inequality leads to $s(\|\mathbf{y} - \mathbf{x}\|_{\mathbf{x}}) \leq \lambda(\mathbf{x})$. Since the equation $\ln(1 + t) = (1 - \lambda(\mathbf{x}))$ has unique solution $t^* > 0$ if $\lambda(\mathbf{x}) < 1$. Moreover,

the function s is strictly increasing and $s(t) < 1$ for $t \geq 0$, which leads to $0 \leq t \leq t^*$. Since $s(\|\mathbf{y} - \mathbf{x}\|_{\mathbf{x}}) \leq \lambda(\mathbf{x})$, we have $\|\mathbf{y} - \mathbf{x}\|_{\mathbf{x}} \leq t^*$. Thus, $\mathcal{L}_F(F(\mathbf{x}))$ is bounded. Hence, $\mathbf{x}^*$ exists due to the well-known Weierstrass theorem.

The uniqueness of $\mathbf{x}^*$ follows from the strict increase of $\omega(\cdot)$. Indeed, for any $\mathbf{x} \in \mathrm{dom}(F)$, by the convexity of g we have $g(\mathbf{x}) - g(\mathbf{x}^*) \geq \mathbf{v}_*^T(\mathbf{x} - \mathbf{x}^*)$, where $\mathbf{v}_* \in \partial g(\mathbf{x}^*)$. By the self-concordant property of f, we also have $f(\mathbf{x}) - f(\mathbf{x}^*) \geq \nabla f(\mathbf{x}^*)^T(\mathbf{x} - \mathbf{x}^*) + \omega(\|\mathbf{x} - \mathbf{x}^*\|_{\mathbf{x}^*})$. Adding these inequalities and using the optimality condition (11), i.e., $0 = \mathbf{v}_* + \nabla f(\mathbf{x}^*)$, we have $F(\mathbf{x}) - F(\mathbf{x}^*) \geq \omega(\|\mathbf{x} - \mathbf{x}^*\|_{\mathbf{x}^*})$. Now, let $\hat{\mathbf{x}}^* \neq \mathbf{x}^*$ is also an optimal solution of (1). We have $0 = F(\hat{\mathbf{x}}^*) - F(\mathbf{x}^*) \geq \omega(\|\mathbf{x} - \mathbf{x}^*\|_{\mathbf{x}^*}) > 0$, which leads to a contradiction. This implies that $\mathbf{x}^* \equiv \hat{\mathbf{x}}^*$. ■

A.2 Proofs of Global Convergence: Theorem 6, Lemma 12 and Theorem 13

In this subsection, we provide the proofs of Theorem 6, Lemma 12 and Theorem 13 in a unified fashion. We first provide a key result quantifying the improvement of the objective as a function of the step-size α_k.

Maximum decrease of the objective function: Let $\beta_k := \|\mathbf{d}^k\|_{\mathbf{H}^k}$, $\lambda_k := \|\mathbf{d}^k\|_{\mathbf{x}^k}$ and:

$$\mathbf{x}^{k+1} := \mathbf{x}^k + \alpha_k \mathbf{d}^k = (1 - \alpha_k)\mathbf{x}^k + \alpha_k \mathbf{s}^k,$$

where $\alpha_k := \frac{\beta_k^2}{\lambda_k(\lambda_k + \beta_k^2)} \in (0, 1]$. We will prove below that the following holds at each iteration of the algorithms:

$$F(\mathbf{x}^{k+1}) \leq F(\mathbf{x}^k) - \omega\left(\frac{\beta_k^2}{\lambda_k}\right). \tag{53}$$

Moreover, the choice of α_k is *optimal* (in the analytical worst-case sense).

Proof Indeed, since g is convex and $\alpha_k \in (0, 1]$, we have $g(\mathbf{x}^{k+1}) = g\big((1 - \alpha_k)\mathbf{x}^k + \alpha_k \mathbf{s}^k\big) \leq (1 - \alpha_k)g(\mathbf{x}^k) + \alpha_k g(\mathbf{s}^k)$, which leads to

$$g(\mathbf{x}^{k+1}) - g(\mathbf{x}^k) \leq \alpha_k(g(\mathbf{s}^k) - g(\mathbf{x}^k)). \tag{54}$$

For $\mathbf{x}^{k+1} \in \mathrm{dom}(F)$ so that $\|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^k} < 1$, the bound (10) holds. Combining (54) with the self-concordant property (10) of f, we obtain

$$\begin{aligned} F(\mathbf{x}^{k+1}) &\leq F(\mathbf{x}^k) + \nabla f(\mathbf{x}^k)^T(\mathbf{x}^{k+1} - \mathbf{x}^k) + \omega_*\big(\|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^k}\big) + \alpha_k\big(g(\mathbf{s}^k) - g(\mathbf{x}^k)\big) \\ &\overset{(16)}{\leq} F(\mathbf{x}^k) + \alpha_k \nabla f(\mathbf{x}^k)^T \mathbf{d}^k + \omega_*\big(\alpha_k \|\mathbf{d}^k\|_{\mathbf{x}^k}\big) + \alpha_k\big(g(\mathbf{s}^k) - g(\mathbf{x}^k)\big). \end{aligned} \tag{55}$$

Since $\mathbf{s}^k$ is the unique solution of (15), by using the optimality condition (17), we get

$$\begin{aligned} -\nabla f(\mathbf{x}^k) - \mathbf{H}_k(\mathbf{s}^k - \mathbf{x}^k) &\in \partial g(\mathbf{s}^k) \Rightarrow \\ -\nabla f(\mathbf{x}^k)^T(\mathbf{s}^k - \mathbf{x}^k) - \|\mathbf{s}^k - \mathbf{x}^k\|_{\mathbf{H}_k}^2 &\in (\mathbf{s}^k - \mathbf{x}^k)^T \partial g(\mathbf{s}^k). \end{aligned} \tag{56}$$

Combining (56) with $g(\mathbf{x}^k) - g(\mathbf{s}^k) \geq \mathbf{v}^T(\mathbf{x}^k - \mathbf{s}^k),\ \mathbf{v} \in \partial(\mathbf{s}^k)$, due to the convexity of $g(\cdot)$, we have

$$g(\mathbf{s}^k) - g(\mathbf{x}^k) \leq -\nabla f(\mathbf{x}^k)^T(\mathbf{s}^k - \mathbf{x}^k) - \|\mathbf{s}^k - \mathbf{x}^k\|_{\mathbf{H}_k}^2. \tag{57}$$

Using (57) in (55) together with the definitions of β_k and λ_k, we obtain

$$F(\mathbf{x}^{k+1}) \overset{(16)}{\leq} F(\mathbf{x}^k) - \alpha_k \beta_k^2 + \omega_* \left(\alpha_k \lambda_k\right). \tag{58}$$

Let us consider the function $\varphi(\alpha) := \alpha\beta_k^2 - \omega_*(\alpha\lambda_k)$. By the definition of $\omega_*(\cdot)$, we can easily show that $\varphi(\alpha)$ attains the maximum at:

$$\alpha_k := \frac{\beta_k^2}{\lambda_k(\lambda_k + \beta_k^2)}, \tag{59}$$

provided that $\alpha_k \in (0, 1]$. We note that the choice of α_k as (59) preserves the condition $\|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^k} < 1$. Moreover, $\varphi(\alpha_k) = \omega(\beta_k^2/\lambda_k)$, which proves (53). ■

Proof of Theorem 6: Since $\mathbf{H}_k := \nabla^2 f(\mathbf{x}^k)$, we observe $\beta_k := \|\mathbf{d}^k\|_{\mathbf{H}_k} \equiv \|\mathbf{d}^k\|_{\mathbf{x}_k} =: \lambda_k$, where $\mathbf{d}^k \equiv \mathbf{d}_n^k$. In this case, the step size α_k in (59) becomes $\alpha_k = \frac{1}{1+\lambda_k}$ which is in $(0, 1)$. Moreover, (53) reduces to:

$$F(\mathbf{x}^{k+1}) \leq F(\mathbf{x}^k) - \omega(\lambda_k),$$

which is indeed (21).

Finally, we assume that, for a given $\sigma \in (0, 1)$, we have $\lambda_k \geq \sigma$ for $0 \leq k \leq k_{\max} - 1$. Since ω strictly increases, it follows from (21) by induction that:

$$F(\mathbf{x}^*) \leq F(\mathbf{x}^k) \leq F(\mathbf{x}^0) - \sum_{j=0}^{k-1} \omega(\lambda_j) \leq F(\mathbf{x}^0) - k\omega(\sigma).$$

This estimate shows that the number of iterations to reach $\lambda_k < \sigma$ is at most $k_{\max} = \left\lfloor \frac{F(\mathbf{x}^0) - F(\mathbf{x}^*)}{\omega(\sigma)} \right\rfloor + 1$. ■

Proof of Lemma 12: Proof of Lemma 12 immediately follows from (53) by taking $\mathbf{H}_k \equiv \mathbf{D}_k$ and $\mathbf{d}^k \equiv \mathbf{d}_g^k$. ■

Proof of Theorem 13: We consider the sequence $\left\{F(\mathbf{x}^k)\right\}_{k\geq 0}$. By Lemma 12, this sequence is nonincreasing. Moreover, $F(\mathbf{x}^0) \geq F(\mathbf{x}^k) \geq F(\mathbf{x}^*)$ for all $k \geq 0$. As a result, the sequence $\left\{F(\mathbf{x}^k)\right\}_{k\geq 0}$ converges to a finite value F^*. By Lemma 12, we can derive

$$\sum_{j=0}^{\infty} \omega \left(\frac{\|\mathbf{d}_g^j\|_{\mathbf{D}_j}^2}{\|\mathbf{d}_g^j\|_{\mathbf{x}^j}} \right) \leq F(\mathbf{x}^0) - F^* < +\infty.$$

Since the function $\omega(\tau) = \tau - \ln(1 + \tau) \geq \frac{\tau^2}{4}$ for $\tau \in (0, 1]$ is increasing, this implies that $\lim_{j\to\infty} \|\mathbf{d}_g^j\|_2^2 / \|\mathbf{d}_g^j\|_{\mathbf{x}^j} = 0$ due to the fact that $\mathbf{D}_k \succeq \underline{L}\mathbb{I} \succ 0$. Since $\mathcal{L}_F(F(\mathbf{x}^0))$ is bounded, by applying Zangwill's convergence theorem in (Zangwill, 1969), we can show that every limit point $\mathbf{x}^*$ of the sequence $\left\{\mathbf{x}^k\right\}_{k\geq 0}$ is the stationary point of (11). Since $\mathbf{x}^*$ is unique, the whole sequence $\left\{\mathbf{x}^k\right\}_{k\geq 0}$ converges to $\mathbf{x}^*$. ■

A.3 Proofs of Local Convergence: Theorem 7, Theorem 11 and Theorem 14

We first provide a fixed-point representation of the optimality conditions and prove some key estimates used in the sequel.

Optimality conditions as fixed-point formulations: Let f be a given standard self-concordant function, g be a given proper, lower semicontinuous and convex function, and $\mathbf{H}_k$ be a given symmetric positive definite matrix. Besides the two key inequalities (9) and (10), we also need the following inequality (Nesterov and Nemirovski, 1994; Nesterov, 2004, Theorem 4.1.6) in the proofs below:

$$(1 - \|\mathbf{y} - \mathbf{x}\|_{\mathbf{x}})^2 \nabla^2 f(\mathbf{x}) \preceq \nabla^2 f(\mathbf{y}) \preceq (1 - \|\mathbf{y} - \mathbf{x}\|_{\mathbf{x}})^{-2} \nabla^2 f(\mathbf{x}), \tag{60}$$

for any $\mathbf{x}, \mathbf{y} \in \mathrm{dom}(f)$ such that $\|\mathbf{y} - \mathbf{x}\|_{\mathbf{x}} < 1$.

Let $\mathbf{x}^*$ be the unique solution of (1) and $\mathbf{x}^*$ be *strongly regular*, i.e., $\nabla^2 f(\mathbf{x}^*) \succ 0$. Then the Dikin ellipsoid $W(\mathbf{x}^*, 1) := \{\mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x} - \mathbf{x}^*\|_{\mathbf{x}^*} < 1\}$ also belongs to $\mathrm{dom}(f)$. Moreover, $\nabla^2 f(\mathbf{x}) \succ 0$ for all $\mathbf{x} \in W(\mathbf{x}^*, 1)$ due to (Nesterov, 2004, Theorem 4.1.5). Hence, the *strong regularity* assumption is sufficient to ensure that $\nabla^2 f$ is positive definite in the neighborhood $W(\mathbf{x}^*, 1)$.

For a fixed $\bar{\mathbf{x}} \in \mathrm{dom}(F)$, where $F := f + g$, we redefined the following operators, based on the fixed-point characterization and (15):

$$P^g_{\bar{\mathbf{x}}}(\mathbf{z}) := P^g_{\nabla^2 f(\bar{\mathbf{x}})}(\mathbf{z}), \quad S_{\bar{\mathbf{x}}}(\mathbf{z}) := \nabla^2 f(\bar{\mathbf{x}})\mathbf{z} - \nabla f(\mathbf{z}), \tag{61}$$

and

$$\mathbf{e}_{\bar{\mathbf{x}}}(\mathbf{H}_k, \mathbf{z}) := \big(\nabla^2 f(\bar{\mathbf{x}}) - \mathbf{H}_k\big)(\mathbf{z} - \mathbf{x}^k). \tag{62}$$

Here, $P^g_{\bar{\mathbf{x}}}$ and $S_{\bar{\mathbf{x}}}$ can be considered as a generalized proximal operator of g and the gradient step of f, respectively. While $\mathbf{e}_{\bar{\mathbf{x}}}(\mathbf{H}_k, \cdot)$ measures the error between $\nabla^2 f(\bar{\mathbf{x}})$ and $\mathbf{H}_k$ along the direction $\mathbf{z} - \mathbf{x}^k$.

Next, given $\mathbf{s}^k$ is the unique solution of (15), we characterize the optimality condition of the original problem (1) and the subproblem (15) based on the $P^g_{\bar{\mathbf{x}}}$, $S_{\bar{\mathbf{x}}}$ and $\mathbf{e}_{\bar{\mathbf{x}}}(\mathbf{H}_k, \cdot)$ operators. From (17), we have

$$S_{\bar{\mathbf{x}}}(\mathbf{x}^k) + \mathbf{e}_{\bar{\mathbf{x}}}(\mathbf{H}_k, \mathbf{s}^k) \in \nabla^2 f(\bar{\mathbf{x}})\mathbf{s}^k + \partial g(\mathbf{s}^k).$$

By the definition of $P^g_{\bar{\mathbf{x}}}$ in (61), the above expression leads to

$$\mathbf{s}^k = P^g_{\bar{\mathbf{x}}}\big(S_{\bar{\mathbf{x}}}(\mathbf{x}^k) + \mathbf{e}_{\bar{\mathbf{x}}}(\mathbf{H}_k, \mathbf{s}^k)\big). \tag{63}$$

By replacing $\bar{\mathbf{x}}$ with $\mathbf{x}^*$, i.e., the unique solution of (1), into (63) we obtain

$$\mathbf{s}^k = P^g_{\mathbf{x}^*}\big(S_{\mathbf{x}^*}(\mathbf{x}^k) + \mathbf{e}_{\mathbf{x}^*}(\mathbf{H}_k, \mathbf{s}^k)\big). \tag{64}$$

Moreover, if we replace $\mathbf{H}_k$ by $\nabla^2 f(\mathbf{x}^*)$ (which is assumed to be positive definite) in the fixed-point expression (12), we finally have

$$\mathbf{x}^* = P^g_{\mathbf{x}^*}\left(S_{\mathbf{x}^*}(\mathbf{x}^*)\right). \tag{65}$$

Formulas (63) to (65) represent the fixed-point formulation of the optimality conditions.

Key estimates: Let $\mathbf{r}_k := \|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*}$ and λ_k be defined by (18). For any $\alpha_k \in (0, 1]$, we prove the following estimates:

$$\|\mathbf{s}_n^{k+1} - \mathbf{s}_n^k\|_{\mathbf{x}^k} \le \frac{\alpha_k^2\lambda_k^2}{1-\alpha_k\lambda_k} + \frac{2\alpha_k\lambda_k - \alpha_k^2\lambda_k^2}{(1-\alpha_k\lambda_k)^2}\|\mathbf{d}^{k+1}\|_{\mathbf{x}^k}, \tag{66}$$

$$\|\mathbf{s}^k - \mathbf{x}^*\|_{\mathbf{x}^*} \le \frac{\mathbf{r}_k^2}{1-\mathbf{r}_k} + \|(\mathbf{H}_k - \nabla^2 f(\mathbf{x}^*))\mathbf{d}^k\|_{\mathbf{x}^*}^*, \tag{67}$$

provided that $\alpha_k\lambda_k < 1$, $\mathbf{r}_k < 1$ and the first estimate (66) requires $\mathbf{H}_k = \nabla^2 f(\mathbf{x}^k)$.

Proof First, by using the nonexpansiveness of $P_{\mathbf{x}^k}^g$ in Lemma 2, it follows from (63) that:

$$\begin{aligned}
\|\mathbf{s}^{k+1} - \mathbf{s}^k\|_{\mathbf{x}^k} &= \Big\|P_{\mathbf{x}^k}^g(S_{\mathbf{x}^k}(\mathbf{x}^{k+1}) + \mathbf{e}_{\mathbf{x}^k}(\mathbf{H}_{k+1}, \mathbf{s}^{k+1})) - P_{\mathbf{x}^k}^g(S_{\mathbf{x}^k}(\mathbf{x}^k) + \mathbf{e}_{\mathbf{x}^k}(\mathbf{H}_k, \mathbf{s}^k))\Big\|_{\mathbf{x}^k} \\
&\overset{(8)}{\le} \Big\|S_{\mathbf{x}^k}(\mathbf{x}^{k+1}) + \mathbf{e}_{\mathbf{x}^k}(\mathbf{H}_k, \mathbf{s}^k) - S_{\mathbf{x}^*}(\mathbf{x}^*)\Big\|_{\mathbf{x}^k}^* \\
&\le \Big\|\nabla f(\mathbf{x}^{k+1}) - \nabla f(\mathbf{x}^k) - \nabla^2 f(\mathbf{x}^k)(\mathbf{x}^{k+1} - \mathbf{x}^k)\Big\|_{\mathbf{x}^k}^* \\
&+ \Big\|\mathbf{e}_{\mathbf{x}^k}(\mathbf{H}_{k+1}, \mathbf{s}^{k+1}) - \mathbf{e}_{\mathbf{x}^k}(\mathbf{H}_k, \mathbf{s}^k)\Big\|_{\mathbf{x}^k}^* \\
&\overset{(i)}{=} \Big\|\int_0^1 \Big(\nabla^2 f(\mathbf{x}^k + \tau(\mathbf{x}^{k+1} - \mathbf{x}^k)) - \nabla^2 f(\mathbf{x}^k)\Big)(\mathbf{x}^{k+1} - \mathbf{x}^k)d\tau\Big\|_{\mathbf{x}^k}^* \\
&+ \Big\|\mathbf{e}_{\mathbf{x}^k}(\mathbf{H}_{k+1}, \mathbf{s}^{k+1}) - \mathbf{e}_{\mathbf{x}^k}(\mathbf{H}_k, \mathbf{s}^k)\Big\|_{\mathbf{x}^k}^*,
\end{aligned} \tag{68}$$

where (i) is due to the mean-value theorem, respectively.

Second, we estimate the first term in (68). For this purpose, we define

$$\begin{aligned}
\mathbf{\Sigma}_k &:= \textstyle\int_0^1 \big(\nabla^2 f(\mathbf{x}^k + \tau(\mathbf{x}^{k+1} - \mathbf{x}^k)) - \nabla^2 f(\mathbf{x}^k)\big)\, d\tau, \\
\mathbf{M}_k &:= \nabla^2 f(\mathbf{x}^k)^{-1/2}\mathbf{\Sigma}_k\nabla^2 f(\mathbf{x}^k)^{-1/2}.
\end{aligned} \tag{69}$$

Based on the proof of (Nesterov, 2004, Theorem 4.1.14), we can show that:

$$\|\mathbf{M}_k\|_2 \le \frac{\|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^k}}{1 - \|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^k}}.$$

Using this estimate, the definition (69) and noting that $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha_k\mathbf{d}^k$, we obtain

$$\begin{aligned}
\|\mathbf{\Sigma}_k(\mathbf{x}^{k+1} - \mathbf{x}^k)\|_{\mathbf{x}^k}^* &= \big[(\mathbf{x}^{k+1} - \mathbf{x}^k)^T\Sigma_k\nabla^2 f(\mathbf{x}^k)^{-1}\Sigma_k(\mathbf{x}^{k+1} - \mathbf{x}^k)\big]^{1/2} \\
&= \big[(\mathbf{x}^{k+1} - \mathbf{x}^k)^T\nabla^2 f(\mathbf{x}^k)^{1/2}\mathbf{M}_k^T\mathbf{M}_k\nabla^2 f(\mathbf{x}^k)^{1/2}(\mathbf{x}^{k+1} - \mathbf{x}^k)\big]^{1/2} \\
&= \|\mathbf{M}_k\nabla^2 f(\mathbf{x}^k)^{1/2}(\mathbf{x}^{k+1} - \mathbf{x}^k)\|_2 \\
&\overset{(i)}{\le} \|\mathbf{M}_k\|_2\big[(\mathbf{x}^{k+1} - \mathbf{x}^k)^T\nabla^2 f(\mathbf{x}^k)(\mathbf{x}^{k+1} - \mathbf{x}^k)\big]^{1/2} \\
&= \|\mathbf{M}_k\|_2\|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^k} \\
&\le \frac{\|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^k}^2}{1 - \|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^k}} \\
&= \frac{\alpha_k^2\|\mathbf{d}^k\|_{\mathbf{x}^k}^2}{1 - \alpha_k\|\mathbf{d}^k\|_{\mathbf{x}^k}},
\end{aligned} \tag{70}$$

where (i) is due to the Cauchy-Schwartz inequality.

Third, we consider the second term in (68) for $\mathbf{H}_k \equiv \nabla^2 f(\mathbf{x}^k)$. By the definition of $\mathbf{e}_{\bar{\mathbf{x}}}$, it is obvious that $\mathbf{e}_{\mathbf{x}^k}(\nabla^2 f(\mathbf{x}^k), \mathbf{s}^k) = 0$. Hence, we have

$$\begin{aligned}\mathcal{T}_2 &:= \left\|\mathbf{e}_{\mathbf{x}^k}(\nabla^2 f(\mathbf{x}^{k+1}), \mathbf{s}^{k+1}) - \mathbf{e}_{\mathbf{x}^k}(\nabla^2 f(\mathbf{x}^k), \mathbf{s}^k)\right\|^*_{\mathbf{x}^k}\\ &= \left\|\mathbf{e}_{\mathbf{x}^k}(\nabla^2 f(\mathbf{x}^{k+1}), \mathbf{s}^{k+1})\right\|^*_{\mathbf{x}^k}\\ &= \left\|\left(\nabla^2 f(\mathbf{x}^{k+1}) - \nabla^2 f(\mathbf{x}^k)\right)\mathbf{d}^{k+1}\right\|^*_{\mathbf{x}^k}.\end{aligned} \tag{71}$$

We now define the following quantity, whose spectral norm we bound below

$$\mathbf{N}_k := \nabla^2 f(\mathbf{x}^k)^{-1/2}\left(\nabla^2 f(\mathbf{x}^{k+1}) - \nabla^2 f(\mathbf{x}^k)\right)\nabla^2 f(\mathbf{x}^k)^{-1/2}. \tag{72}$$

By applying (60) with $\mathbf{x} = \mathbf{x}^k$ and $\mathbf{y} = \mathbf{x}^{k+1}$, we can bound the spectral norm of $\mathbf{N}_k$ as follows:

$$\begin{aligned}\|\mathbf{N}_k\|_2 &\le \max\left\{1 - \left(1 - \|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^k}\right)^2, \left(1 - \|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^k}\right)^{-2} - 1\right\}\\ &= \frac{2\|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^k} - \|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2_{\mathbf{x}^k}}{(1 - \|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^k})^2}.\end{aligned} \tag{73}$$

Therefore, from (71) we can obtain the following estimate:

$$\begin{aligned}(\mathcal{T}_2)^2 &= \mathbf{e}_{\mathbf{x}^k}(\nabla^2 f(\mathbf{x}^{k+1}), \mathbf{s}^{k+1})^T \nabla^2 f(\mathbf{x}^k)^{-1} \mathbf{e}_{\mathbf{x}^k}(\nabla^2 f(\mathbf{x}^{k+1}), \mathbf{s}^{k+1})\\ &= (\mathbf{d}^{k+1})^T\ \nabla^2 f(\mathbf{x}^k)^{1/2}\ \mathbf{N}_k^2\ \nabla^2 f(\mathbf{x}^k)^{1/2}\ \mathbf{d}^{k+1}\\ &\le \|\mathbf{N}_k\|_2^2\, \|\mathbf{d}^{k+1}\|^2_{\mathbf{x}^k}.\end{aligned} \tag{74}$$

By substituting (73) into (74) and noting that $\alpha_k \mathbf{d}^k = \mathbf{x}^{k+1} - \mathbf{x}^k$, we obtain

$$\mathcal{T}_2 \le \frac{2\alpha_k \left\|\mathbf{d}^k\right\|_{\mathbf{x}^k} - \alpha_k^2 \left\|\mathbf{d}^k\right\|^2_{\mathbf{x}^k}}{(1 - \alpha_k \left\|\mathbf{d}^k\right\|_{\mathbf{x}^k})^2}\|\mathbf{d}^{k+1}\|_{\mathbf{x}^k}. \tag{75}$$

Now, by substituting (70) and (75) into (68) and noting that $\mathbf{H}_k \equiv \nabla^2 f(\mathbf{x}^k)$, $\mathbf{s}^k \equiv \mathbf{s}_n^k$, $\mathbf{d}^k \equiv \mathbf{d}_n^k$ and $\lambda_k \equiv \left\|\mathbf{d}_n^k\right\|_{\mathbf{x}^k}$, we obtain

$$\left\|\mathbf{s}_n^{k+1} - \mathbf{s}_n^k\right\|_{\mathbf{x}^k} \le \frac{\alpha_k^2 \left\|\mathbf{d}_n^k\right\|^2_{\mathbf{x}^k}}{1 - \alpha_k \left\|\mathbf{d}_n^k\right\|_{\mathbf{x}^k}} + \frac{2\alpha_k \left\|\mathbf{d}_n^k\right\|_{\mathbf{x}^k} - \alpha_k^2 \left\|\mathbf{d}_n^k\right\|^2_{\mathbf{x}^k}}{(1 - \alpha_k \left\|\mathbf{d}_n^k\right\|_{\mathbf{x}^k})^2}\|\mathbf{d}_n^{k+1}\|_{\mathbf{x}^k}.$$

which is indeed (66).

Similarly to the proof of (68) and (70), we have

$$\begin{aligned}\|\mathbf{s}^k - \mathbf{x}^*\|_{\mathbf{x}^*} &\overset{(65)}{=} \left\|P^g_{\mathbf{x}^*}(S_{\mathbf{x}^*}(\mathbf{x}^k) + \mathbf{e}_{\mathbf{x}^*}(\mathbf{H}_k, \mathbf{s}^k)) - P^g_{\mathbf{x}^*}(S_{\mathbf{x}^*}(\mathbf{x}^*))\right\|_{\mathbf{x}^*}\\ &\overset{(8)}{\le} \left\|S_{\mathbf{x}^*}(\mathbf{x}^k) + \mathbf{e}_{\mathbf{x}^*}(\mathbf{H}_k, \mathbf{s}^k) - S_{\mathbf{x}^*}(\mathbf{x}^*)\right\|^*_{\mathbf{x}^*}\\ &\le \left\|\nabla f(\mathbf{x}^k) - \nabla f(\mathbf{x}^*) - \nabla^2 f(\mathbf{x}^*)(\mathbf{x}^k - \mathbf{x}^*)\right\|^*_{\mathbf{x}^*} + \|\mathbf{e}_{\mathbf{x}^*}(\mathbf{H}_k, \mathbf{s}^k)\|^*_{\mathbf{x}^*}\\ &= \left\|\int_0^1 \left(\nabla^2 f(\mathbf{x}^* + \tau(\mathbf{x}^k - \mathbf{x}^*)) - \nabla^2 f(\mathbf{x}^*)\right)(\mathbf{x}^k - \mathbf{x}^*)d\tau\right\|^*_{\mathbf{x}^*} + \|\mathbf{e}_{\mathbf{x}^*}(\mathbf{H}_k, \mathbf{s}^k)\|^*_{\mathbf{x}^*}\\ &\overset{(70)}{\le} \frac{\|\mathbf{x}^k - \mathbf{x}^*\|^2_{\mathbf{x}^*}}{1 - \|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*}} + \left\|\left(\mathbf{H}_k - \nabla^2 f(\mathbf{x}^*)\right)\mathbf{d}^k\right\|^*_{\mathbf{x}^*},\end{aligned} \tag{76}$$

which is indeed (67) since $\mathbf{r}_k = \|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*}$. ■

Proof of Theorem 7: Since $\mathbf{x}^k = \mathbf{s}_n^k - \mathbf{d}_n^k$ due to (20), we have $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha_k \mathbf{d}_n^k = \mathbf{s}_n^k - (1-\alpha_k)\mathbf{d}_n^k$, which leads to

$$\mathbf{d}_n^{k+1} = \mathbf{s}_n^{k+1} - \mathbf{x}^{k+1} = \mathbf{s}_n^{k+1} - \mathbf{s}_n^k + (1-\alpha_k)\mathbf{d}_n^k.$$

By applying the triangle inequality to the above expression, we have

$$\|\mathbf{d}_n^{k+1}\|_{\mathbf{x}^k} = \|\mathbf{s}_n^{k+1} - \mathbf{s}_n^k + (1-\alpha_k)\mathbf{d}_n^k\|_{\mathbf{x}^k} \leq \|\mathbf{s}_n^{k+1} - \mathbf{s}_n^k\|_{\mathbf{x}^k} + (1-\alpha_k)\|\mathbf{d}_n^k\|_{\mathbf{x}^k}. \tag{77}$$

Substituting (66) into (77) we obtain

$$\|\mathbf{d}_n^{k+1}\|_{\mathbf{x}^k} \leq \frac{\alpha_k^2\lambda_k^2}{1-\alpha_k\lambda_k} + \frac{2\alpha_k\lambda_k - \alpha_k^2\lambda_k^2}{(1-\alpha_k\lambda_k)^2}\|\mathbf{d}^{k+1}\|_{\mathbf{x}^k} + (1-\alpha_k)\lambda_k.$$

Rearranging this inequality we get

$$\|\mathbf{d}_n^{k+1}\|_{\mathbf{x}^k} \leq \left(\frac{(1-\alpha_k\lambda_k)\left(1-\alpha_k+(2\alpha_k^2-\alpha_k)\lambda_k\right)}{1-4\alpha_k\lambda_k+2\alpha_k^2\lambda_k^2}\right)\lambda_k, \tag{78}$$

provided that $1 - 4\alpha_k\lambda_k + 2\alpha_k^2\lambda_k^2 > 0$. Now, by applying (60) with $\mathbf{x} = \mathbf{x}^k$ and $\mathbf{y} = \mathbf{x}^{k+1}$, one can show that

$$\|\mathbf{d}_n^{k+1}\|_{\mathbf{x}^{k+1}} \leq \frac{\|\mathbf{d}_n^{k+1}\|_{\mathbf{x}^k}}{1-\alpha_k\|\mathbf{d}_n^k\|_{\mathbf{x}^k}}. \tag{79}$$

We note that $1 - 4\alpha_k\lambda_k + 2\alpha_k^2\lambda_k^2 > 0$ if $\alpha_k\lambda_k < 1 - 1/\sqrt{2}$. By combining (78) and (79) we obtain

$$\|\mathbf{d}_n^{k+1}\|_{\mathbf{x}^{k+1}} \leq \left(\frac{1-\alpha_k+(2\alpha_k^2-\alpha_k)\lambda_k}{1-4\alpha_k\lambda_k+2\alpha_k^2\lambda_k^2}\right)\lambda_k,$$

which is (22).

Next, we consider the sequence $\{\mathbf{x}^k\}_{k\geq 0}$ generated by damped step proximal Newton method (20) with the step size $\alpha_k = (1+\lambda_k)^{-1}$. It is clear that (22) is transformed into:

$$\lambda_{k+1} \leq \frac{2\lambda_k}{1-2\lambda_k-\lambda_k^2}\lambda_k. \tag{80}$$

Assuming $\lambda_k \leq \bar{\sigma} := \sqrt{5}-2$, we can easily deduce that $\frac{2\lambda_k}{1-2\lambda_k-\lambda_k^2} \leq 1$ and thus, $\lambda_{k+1} \leq \lambda_k$. By induction, if $\lambda_0 \leq \bar{\sigma}$ then, $\lambda_{k+1} \leq \lambda_k$ for all $k \geq 0$. Moreover, we have $\lambda_{k+1} \leq \frac{2}{1-2\bar{\sigma}-\bar{\sigma}^2}\lambda_k^2$, which shows that the sequence $\{\lambda_k\}_{k\geq 0}$ converges to zero at a quadratic rate, which completes the proof of part b).

Now, since $\alpha_k = 1$, the estimate (22) reduces to $\lambda_{k+1} \leq \frac{\lambda_k^2}{1-4\lambda_k+2\lambda_k^2}$. By the same argument as in the proof of part b), we can show that the sequence $\{\lambda_k\}_{k\geq 0}$ converges to zero at a quadratic rate.

Finally, we prove the last statement in Theorem 7. By substituting $\mathbf{H}_k := \nabla^2 f(\mathbf{x}^k)$ into (66), we obtain

$$\|\mathbf{s}^k - \mathbf{x}^*\|_{\mathbf{x}^*} \le \frac{\mathbf{r}_k^2}{1-\mathbf{r}_k} + \|(\nabla^2 f(\mathbf{x}^k) - \nabla^2 f(\mathbf{x}^*))\mathbf{d}^k\|_{\mathbf{x}^*}^*. \tag{81}$$

Let $\mathcal{T}_3 := \|(\nabla^2 f(\mathbf{x}^k) - \nabla^2 f(\mathbf{x}^*))\mathbf{d}^k\|_{\mathbf{x}^*}^*$. Similarly to the proof of (75), we can show that:

$$\mathcal{T}_3 \le \left[\frac{2\|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*} - \|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*}^2}{(1-\|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*})^2}\right]\|\mathbf{d}^k\|_{\mathbf{x}^*} \le \alpha_k \frac{(2-r_k)r_k}{(1-r_k)^2}(r_{k+1}+r_k). \tag{82}$$

Here the second inequality follows from the fact that $\|\mathbf{d}^k\|_{\mathbf{x}^*} = \alpha_k\|\mathbf{x}^{k+1}-\mathbf{x}^k\|_{\mathbf{x}^*} \le \alpha_k[\|\mathbf{x}^{k+1}-\mathbf{x}^*\|_{\mathbf{x}^*} + \|\mathbf{x}^k-\mathbf{x}^*\|_{\mathbf{x}^*}] = \alpha_k(r_{k+1}+r_k)$. We also have $r_{k+1} = \|\mathbf{x}^{k+1}-\mathbf{x}^*\|_{\mathbf{x}^*} = \|(1-\alpha_k)\mathbf{x}^k + \alpha_k\mathbf{s}^k - \mathbf{x}^*\|_{\mathbf{x}^*} \le (1-\alpha_k)r_k + \alpha_k\|\mathbf{s}_k - \mathbf{x}^*\|_{\mathbf{x}^*}$. Using these inequalities, (82) and (81) we get

$$r_{k+1} \le (1-\alpha_k)r_k + \alpha_k\frac{r_k^2}{1-r_k} + \alpha_k^2\frac{(2-r_k)r_k}{(1-r_k)^2}(r_{k+1}+r_k). \tag{83}$$

Rearranging this inequality to obtain

$$r_{k+1} \le \left(\frac{1-\alpha_k + (2\alpha_k^2+3\alpha_k-2)r_k + (1-\alpha_k-\alpha_k^2)r_k^2}{1-2(1+\alpha_k^2)r_k + (1+\alpha_k^2)r_k^2}\right)r_k. \tag{84}$$

We consider two cases:

Case 1: $\alpha_k = 1$: We have $r_{k+1} \le \frac{3-r_k}{1-4r_k+2r_k^2}r_k^2$. Hence, if $r_k < 1-1/\sqrt{2}$ then $1-4r_k+2r_k^2 > 0$. Moreover, $r_{k+1} \le r_k$ if $3r_k - r_k^2 < 1 - 4r_k + 2r_k^2$, which is satisfied if $r_k < (7-\sqrt{37})/6 \approx 0.152873$. Now, if we assume that $r_0 \le \sigma \in (0, (7-\sqrt{37})/6)$, then, by induction, we have $r_{k+1} \le \frac{3-\sigma}{1-4\sigma+2\sigma^2}r_k^2$. This shows that $\{r_k\}_{k\ge 0}$ locally converges to 0^+ at a quadratic rate. Since $r_k := \left\|\mathbf{x}^k - \mathbf{x}^*\right\|_{\mathbf{x}^*}$, we can conclude that $\mathbf{x}^k \to \mathbf{x}^*$ at a quadratic rate as $k \to \infty$.

Case 2: $\alpha_k = (1+\lambda_k)^{-1}$: Since $\lambda_k = \left\|\mathbf{x}^{k+1} - \mathbf{x}^k\right\|_{\mathbf{x}^k} \le \frac{\left\|\mathbf{x}^{k+1}-\mathbf{x}^*\right\|_{\mathbf{x}^*} + \left\|\mathbf{x}^k-\mathbf{x}^*\right\|_{\mathbf{x}^*}}{1-\left\|\mathbf{x}^k-\mathbf{x}^*\right\|_{\mathbf{x}^*}} = \frac{r_{k+1}+r_k}{1-r_k}$. We have $1-\alpha_k \le \frac{r_{k+1}+r_k}{(1+\lambda_k)(1-r_k)} \le \frac{r_{k+1}+r_k}{1-r_k}$. Substituting this into (83) and using the fact that $\alpha_k \le 1$, we have

$$r_{k+1} \le \frac{(r_{k+1}+r_k)r_k}{1-r_k} + \frac{r_k^2}{1-r_k} + \frac{(2-r_k)r_k}{(1-r_k)^2}(r_{k+1}+r_k).$$

Rearranging this inequality, we finally get

$$r_{k+1} \le \frac{4-3r_k}{1-5r_k+3r_k^2}r_k^2. \tag{85}$$

Since $1-5r_k+3r_k^2 > 0$ if $r_k < (5-\sqrt{13})/6$, we can see from (85) that $r_k < (9-\sqrt{57})/12 \approx 0.120847$ then $r_{k+1} \le r_k$. By induction, if we choose $r_0 \le \bar{\sigma} \in (0, (9-\sqrt{57})/12)$ then $r_{k+1} \le \frac{4-3\bar{\sigma}}{1-5\bar{\sigma}+3\bar{\sigma}^2}r_k^2$, which shows that $\{r_k\}_{k\ge 0}$ converges to 0^+ at a quadratic rate. Consequently, the sequence $\left\{\mathbf{x}^k\right\}_{k\ge 0}$ locally converges to $\mathbf{x}^*$ at a quadratic rate. ■

Proof of Theorem 11: We first prove the statement (a). Since $\mathbf{x}^{k+1} \equiv \mathbf{s}_q^k$ due to (25), from (67) we have

$$\mathbf{r}_{k+1} \leq \frac{\mathbf{r}_k^2}{1-\mathbf{r}_k} + \left\| \left(\mathbf{H}_k - \nabla^2 f(\mathbf{x}^*)\right)(\mathbf{x}^{k+1} - \mathbf{x}^k)\right\|_{\mathbf{x}^*}^*. \tag{86}$$

Now, by using the condition (26), we can easily show that the sequence $\left\{\mathbf{x}^k\right\}_{k\geq 0}$ converges super-linearly to $\mathbf{x}^*$ provided that $\left\|\mathbf{x}^0 - \mathbf{x}^*\right\|_{\mathbf{x}^*} \leq \rho_0 < 1$.

Next, we prove the statement (b). It is well-known (see, e.g., Nocedal and Wright (2006)) that if matrix $\mathbf{H}_k$ is positive definite and $(\mathbf{y}^k)^T(\mathbf{z}^k) > 0$ then the matrix $\mathbf{H}_{k+1}$ updated by (24) is also positive definite. Indeed, we have $(\mathbf{y}^k)^T(\mathbf{z}^k) = \int_0^1 (\mathbf{z}^k)^T \nabla^2 f(\mathbf{x}^k + t\mathbf{z}^k)\mathbf{z}^k dt$. Therefore, under the condition $\left\|\mathbf{z}^k\right\|_{\mathbf{x}^k} < 1$, we can show that $(\mathbf{y}^k)^T(\mathbf{z}^k) \geq (\mathbf{z}^k)^T\nabla^2 f(\mathbf{x}^k)\mathbf{z}^k = \left\|\mathbf{z}^k\right\|_{\mathbf{x}^k}^2 > 0$. By multiplying (24) by $\mathbf{z}^k$ we can easily see that $\mathbf{H}_{k+1}$ satisfies the secant equation (23).

Finally, we estimate $\left\|\mathbf{y}^k - \nabla^2 f(\mathbf{x}^*)\mathbf{z}^k\right\|_{\mathbf{x}^*}^*$ as follows:

$$\|\mathbf{y}^k - \nabla^2 f(\mathbf{x}^*)\mathbf{z}^k\|_{\mathbf{x}^*}^* \leq \frac{\mathbf{r}_k + \mathbf{r}_{k+1}}{(1-\mathbf{r}_k)(1-\mathbf{r}_{k+1})}\|\mathbf{z}^k\|_{\mathbf{x}^*}. \tag{87}$$

Now, by assumption that $\sum_{k=0}^{\infty}\mathbf{r}_k < +\infty$, we obtain from (87) that $\sum_{k=0}^{\infty}\varepsilon_k < +\infty$, where $\varepsilon_k := \frac{\mathbf{r}_k+\mathbf{r}_{k+1}}{(1-\mathbf{r}_k)(1-\mathbf{r}_{k+1})}$. By applying (Byrd and Nocedal, 1989, Theorem 3.2.), we can show that the Dennis-Moré condition (26) is satisfied. This implies that the sequence $\left\{\mathbf{x}^k\right\}_{k\geq 0}$ generated by scheme (25) converges super-linearly to $\mathbf{x}^*$. ■

Proof of Theorem 14: For $\left\|\mathbf{x}^k - \mathbf{x}^*\right\|_{\mathbf{x}^*} < 1$, from (67), we have

$$\|\mathbf{s}_g^k - \mathbf{x}^*\|_{\mathbf{x}^*} \leq \frac{\|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*}^2}{1 - \|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*}} + \left\|\left(\mathbf{D}_k - \nabla^2 f(\mathbf{x}^*)\right)\mathbf{d}^k\right\|_{\mathbf{x}^*}^*. \tag{88}$$

Now, using the condition $\left\|\left(\mathbf{D}_k - \nabla^2 f(\mathbf{x}^*)\right)\mathbf{d}^k\right\|_{\mathbf{x}^*}^* \leq (1/2)\|\mathbf{d}_g^k\|_{\mathbf{x}^*}$, (88) implies:

$$\begin{aligned}\|\mathbf{s}_g^k - \mathbf{x}^*\|_{\mathbf{x}^*} &\leq \frac{\|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*}^2}{1 - \|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*}} + \gamma\|\mathbf{d}_g^k\|_{\mathbf{x}^*} \\ &\leq \frac{\|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*}^2}{1 - \|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*}} + \gamma\|\mathbf{s}_g^k - \mathbf{x}^*\|_{\mathbf{x}^*} + \gamma\|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*},\end{aligned}$$

where $\gamma \in (0, 1/2)$. Rearranging this inequality, we obtain

$$\|\mathbf{s}_g^k - \mathbf{x}^*\|_{\mathbf{x}^*} \leq \frac{1}{1-\gamma}\left(\gamma + \frac{\|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*}}{1 - \|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*}}\right)\|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*}. \tag{89}$$

Now, since $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha_k\mathbf{d}_g^k = (1-\alpha_k)\mathbf{x}^k + \alpha_k\mathbf{s}_g^k$, we can further estimate from (89) as

$$\begin{aligned}\|\mathbf{x}^{k+1} - \mathbf{x}^*\|_{\mathbf{x}^*} &\leq (1-\alpha_k)\|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*} + \alpha_k\|\mathbf{s}_g^k - \mathbf{x}^*\|_{\mathbf{x}^*} \\ &\leq \left[1 - \alpha_k + \frac{\alpha_k}{1-\gamma}\left(\gamma + \frac{\|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*}}{1 - \|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*}}\right)\right]\|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*}.\end{aligned} \tag{90}$$

Let us define $\tilde{\psi}_k := (1-\alpha_k) + \frac{\alpha_k}{1-\gamma}\left(\gamma + \frac{\|\mathbf{x}^k-\mathbf{x}^*\|_{\mathbf{x}^*}}{1-\|\mathbf{x}^k-\mathbf{x}^*\|_{\mathbf{x}^*}}\right)$. Then, for $\gamma < 1/2$, $\tilde{\psi}_k < 1$ if $\|\mathbf{x}^k-\mathbf{x}^*\|_{\mathbf{x}^*} < \frac{1-2\gamma}{2(1-\gamma)}$. Therefore, by induction, if we choose $\|\mathbf{x}^0-\mathbf{x}^*\|_{\mathbf{x}^*} < \frac{1-2\gamma}{2(1-\gamma)}$, then $\|\mathbf{x}^k-\mathbf{x}^*\|_{\mathbf{x}^*} < \frac{1-2\gamma}{2(1-\gamma)}$ for all $k \geq 0$. Moreover, $\|\mathbf{x}^{k+1}-\mathbf{x}^*\|_{\mathbf{x}^*} \leq \tilde{\psi}_k\|\mathbf{x}^k-\mathbf{x}^*\|_{\mathbf{x}^*}$ for $k \geq 0$ and $\tilde{\psi}_k \in [0,1)$. This implies that $\left\{\|\mathbf{x}^k-\mathbf{x}^*\|_{\mathbf{x}^*}\right\}_{k\geq 0}$ linearly converges to zero with the factor $\tilde{\psi}_k$.

Finally, we assume that $\mathbf{D}_k := L_k\mathbb{I}$, the quantity in (72) satisfies

$$\mathbf{N}_* := \nabla^2 f(\mathbf{x}^*)^{-1/2}\left(\nabla^2 f(\mathbf{x}^*) - \mathbf{H}_k\right)\nabla^2 f(\mathbf{x}^*)^{-1/2} = \mathbb{I} - L_k\nabla^2 f(\mathbf{x}^*)^{-1}.$$

Then, we can easily observe that:

$$\|\mathbf{N}_*\|_2 = \left\|\mathbb{I} - L_k\nabla^2 f(\mathbf{x}^*)^{-1}\right\|_2 \leq \max\left\{\left|1-\frac{L_k}{\sigma^*_{\min}}\right|, \left|1-\frac{L_k}{\sigma^*_{\max}}\right|\right\} := \gamma_*, \tag{91}$$

where $\sigma^*_{\min}$ (respectively, $\sigma^*_{\max}$) is the smallest (respectively, largest) eigenvalue of $\nabla^2 f(\mathbf{x}^*)$. Using the estimate (91), we can derive

$$\left\|\left(\mathbf{D}_k - \nabla^2 f(\mathbf{x}^*)\right)\mathbf{d}_g^k\right\|^*_{\mathbf{x}^*} \overset{(91)}{\leq} \|\mathbf{N}_*\|_2\|\mathbf{s}^k-\mathbf{x}^k\|_{\mathbf{x}^*} \leq \gamma_*\|\mathbf{d}_g^k\|_{\mathbf{x}^*},$$

which proves the last conclusion of Theorem 14. ■

References

K. Banaszek, G. M. D'Ariano, M. G. A. Paris, and M. F. Sacchi. Maximum-likelihood estimation of the density matrix. *Phys. Rev. A.*, 61(010304):1–4, 1999.

O. Banerjee, L. El Ghaoui, and A. d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *The Journal of Machine Learning Research*, 9:485–516, 2008.

A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sciences*, 2(1):183–202, 2009a.

A. Beck and M. Teboulle. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *IEEE Trans Image Process.*, 18(11):2419–2434, 2009b.

S. Becker and M.J. Fadili. A quasi-Newton proximal splitting method. In *Proceedings of Neutral Information Processing Systems Foundation*, 2012.

S. Becker, E. Candès, and M. Grant. Templates for convex cone problems with applications to sparse signal recovery. *Mathematical Programming Computation*, 3:165–218, 2011. ISSN 1867-2949.

A. Ben-Tal and A.K. Nemirovski. *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*. SIAM, 2001.

D.P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Prentice Hall, 1989.

J.F. Bonnans. Local analysis of Newton-type methods for variational inequalities and nonlinear programming. *Appl. Math. Optim*, 29:161–186, 1994.

S. Boyd and L. Vandenberghe. *Convex Optimization*. University Press, Cambridge, 2004.

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

L.M. Briceno-Arias and P.L. Combettes. A monotone + skew splitting model for composite monotone inclusions in duality. *SIAM J. Optim.*, 21(4):1230–1250, 2011.

R. H. Byrd and J. Nocedal. A tool for the analysis of quasi-Newton methods with application to unconstrained minimization. *SIAM J. Numer. Anal.*, 26(3):727–739, 1989.

E. Candes and T. Tao. The Dantzig selector: Statistical estimation when p is much larger than n. *Annals of Statistics*, 35(6):2313–2351, 2007.

A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.

E. Chouzenoux, J.-C. Pesquet, and A. Repetti. Variable metric forward-backward algorithm for minimizing the sum of a differentiable function and a convex function. *J. Optim. Theory Appl.*, DOI 10.1007/s10957-013-0465-7:1–22, 2013.

P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Model. Simul.*, 4:1168–1200, 2005.

A. S. Dalalyan, M. Hebiri, K. Meziani, and J. Salmon. Learning heteroscedastic models by convex programming under group sparsity. *Proc. of the International conference on Machine Learning*, pages 1–8, 2013.

A. P. Dempster. Covariance selection. *Biometrics*, 28:157–175, 1972.

J.E. Dennis and J. J. Moré. A characterisation of superlinear convergence and its application to quasi–Newton methods. *Mathemathics of Computation*, 28:549–560, 1974.

J. Eckstein and D. Bertsekas. On the Douglas - Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Math. Program.*, 55:293–318, 1992.

F. Facchinei and J.-S. Pang. *Finite-Dimensional Variational Inequalities and Complementarity Problems*, volume 1-2. Springer-Verlag, 2003.

D. Goldfarb and S. Ma. Fast alternating linearization methods of minimization of the sum of two convex functions. *Math. Program., Ser. A*, pages 1–34, 2012.

T. Goldstein and S. Osher. The split Bregman method for ℓ_1-pegularized problems. *SIAM J. Imaging Sciences*, 2(2):323–343, 2009.

T. Goldstein, B. ODonoghue, and S. Setzer. Fast alternating direction optimization methods. Tech. report., Department of Mathematics, University of California, Los Angeles, USA, May 2012.

M. Grant, S. Boyd, and Y. Ye. Disciplined convex programming. In L. Liberti and N. Maculan, editors, *Global Optimization: From Theory to Implementation*, Nonconvex Optimization and its Applications, pages 155–210. Springer, 2006.

Z.T. Harmany, R.F. Marcia, and R. M. Willett. This is SPIRAL-TAP: Sparse poisson intensity reconstruction algorithms - theory and practice,. *IEEE Transactions on Image Processing*, 21(3):1084–1096, 2012.

Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal. *Fundamentals of Convex Analysis*. Springer, 2001.

C. J. Hsieh, M.A. Sustik, I.S. Dhillon, and P. Ravikumar. Sparse inverse covariance matrix estimation using quadratic approximation. *Advances in Neutral Information Processing Systems (NIPS)*, 24:1–18, 2011.

J. Kim and H. Park. Fast active-set-type algorithms for ℓ_1-regularized linear regression. In *Proceedings of the 13th International Conference on Artificial Intelligience and Statistics (AISTATS)*, volume 9, pages 397–404, Sardinia, Italy, 2010.

A. Kyrillidis and V. Cevher. Fast proximal algorithms for self-concordant function minimization with application to sparse graph selection. *Proc. of the 38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1–5, 2013.

A. Kyrillidis, R. Karimi Mahabadi, Q. Tran-Dinh, and V. Cevher. Scalable sparse covariance estimation via self-concordance. In *Proc. of the 28th International Conference on Artificial Intelligence (AAAI-14)*, pages 1–9. 2014.

J.D. Lee, Y. Sun, and M.A. Saunders. Proximal Newton-type methods for convex optimization. *Advances in Neural Information Processing Systems (NIPS)*, 25:827–835, 2012.

J. Löfberg. YALMIP: A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

Z. Lu. Adaptive first-order methods for general sparse inverse covariance selection. *SIAM Journal on Matrix Analysis and Applications*, 31(4):2000–2016, 2010.

H. Mine and M. Fukushima. A minimization method for the sum of a convex function and a continuously differentiable function. *J. Optim. Theory Appl.*, 33:9–23, 1981.

A.S. Nemirovskii and M.J. Todd. Interior-point methods for optimization. *Acta Numerica*, pages 191–234, 2008.

Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*, volume 87 of *Applied Optimization*. Kluwer Academic Publishers, 2004.

Y. Nesterov. Smooth minimization of non-smooth functions. *Math. Program.*, 103(1):127–152, 2005a.

Y. Nesterov. Excessive gap technique in nonsmooth convex minimization. *SIAM J. Optimization*, 16(1):235–249, 2005b.

Y. Nesterov. Gradient methods for minimizing composite objective function. *Math. Program.*, 140(1):125–161, 2007.

Y. Nesterov. Barrier subgradient method. *Math. Program., Ser. B*, 127:31–56, 2011.

Y. Nesterov and A. Nemirovski. *Interior-point Polynomial Algorithms in Convex Programming*. Society for Industrial Mathematics, 1994.

Y. Nesterov and M.J. Todd. Self-scaled barriers and interior-point methods for convex programming. *Math. Oper. Research*, 22(1):1–42, 1997.

J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, 2 edition, 2006.

P.A. Olsen, F. Oztoprak, J. Nocedal, and S.J. Rennie. Newton-like methods for sparse inverse covariance estimation. *Advances in Neural Information Processing Systems (NIPS)*, pages 1–9, 2012.

P. Ravikumar, M. J. Wainwright, G. Raskutti, and B. Yu. High-dimensional covariance estimation by minimizing ℓ_1-penalized log-determinant divergence. *Electron. J. Statist.*, 5:935–988, 2011.

S. M. Robinson. Strongly Regular Generalized Equations. *Mathematics of Operations Research, Vol. 5, No. 1 (Feb., 1980), pp. 43-62*, 5:43–62, 1980.

R. T. Rockafellar. *Convex Analysis*, volume 28 of *Princeton Mathematics Series*. Princeton University Press, 1970.

R. T. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM J. Control Opt.*, 14:877–898, 1976.

B. Rolfs, B. Rajaratnam, D. Guillot, I. Wong, and A. Maleki. Iterative thresholding algorithm for sparse inverse covariance estimation. In *Advances in Neural Information Processing Systems 25*, pages 1583–1591, 2012.

K. Scheinberg and I. Rish. SINCO-a greedy coordinate ascent method for sparse inverse covariance selection problem. *Tech. Report*, IBM RC24837:1–21, 2009.

K. Scheinberg, S. Ma, and D. Goldfarb. Sparse inverse covariance selection via alternating linearization methods. *Neural Information Processing Systems (NIPS)*, pages 1–9, 2010.

M. Schmidt, N.L. Roux, and F. Bach. Convergence rates of inexact proximal-gradient methods for convex optimization. *Neural Information Processing Systems (NIPS)*, 2011.

N. Städler, P. Bülmann, and S. Van de Geer. ℓ_1-Penalization for Mixture Regression Models. *Tech. Report.*, pages 1–35, 2012.

Q. Tran-Dinh, A. Kyrillidis, and V. Cevher. A proximal newton framework for composite minimization: Graph learning without Cholesky decompositions and matrix inversions. *International Conference on Machine Learning (ICML)*, 28(2):271–279, 2013a.

Q. Tran-Dinh, I. Necoara, C. Savorgnan, and M. Diehl. An inexact perturbed path-following method for Lagrangian decomposition in large-scale separable convex optimization. *SIAM J. Optim.*, 23(1):95–125, 2013b.

Q. Tran-Dinh, C. Savorgnan, and M. Diehl. Combining Lagrangian decomposition and excessive gap smoothing technique for solving large-scale separable convex optimization problems. *Compt. Optim. Appl.*, 55(1):75–111, 2013c.

Q. Tran-Dinh, A. Kyrillidis, and V. Cevher. An inexact proximal path-following algorithm for constrained convex minimization. *SIAM J. Optimization (accepted)*, 2014a.

Q. Tran-Dinh, Y. H. Li, and V. Cevher. Barrier smoothing for nonsmooth convex minimization. In *Proc. of the 2014 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1–4, 2014b.

E. van den Berg and M. P. Friedlander. Probing the pareto frontier for basis pursuit solutions. *SIAM J. Sci. Comput.*, 31(2):890–912, 2008.

S. J. Wright, R. Nowak, and M. Figueiredo. Sparse reconstruction by separable approximation. *IEEE Trans. Signal Processing*, 57:2479–2493, 2009.

X. Yuan. Alternating direction method for covariance selection models. *Journal of Scientific Computing*, 51(2):261–273, 2012.

W.I. Zangwill. *Nonlinear Programming.* Prentice Hall, 1969.

Journal of Machine Learning Research 16 (2015) 417-453 Submitted 9/13; Revised 5/14; Published 3/15

Network Granger Causality with Inherent Grouping Structure

Sumanta Basu SUMBOSE@UMICH.EDU
Department of Statistics
University of Michigan
Ann Arbor, MI 48109-1092, USA

Ali Shojaie ASHOJAIE@U.WASHINGTON.EDU
Department of Biostatistics
University of Washington
Seattle, WA, USA

George Michailidis GMICHAIL@UMICH.EDU
Department of Statistics
University of Michigan
Ann Arbor, MI 48109-1092, USA

Editor: Bin Yu

Abstract

The problem of estimating high-dimensional network models arises naturally in the analysis of many biological and socio-economic systems. In this work, we aim to learn a network structure from temporal panel data, employing the framework of Granger causal models under the assumptions of sparsity of its edges and inherent grouping structure among its nodes. To that end, we introduce a group lasso regression regularization framework, and also examine a thresholded variant to address the issue of group misspecification. Further, the norm consistency and variable selection consistency of the estimates are established, the latter under the novel concept of direction consistency. The performance of the proposed methodology is assessed through an extensive set of simulation studies and comparisons with existing techniques. The study is illustrated on two motivating examples coming from functional genomics and financial econometrics.

Keywords: Granger causality, high dimensional networks, panel vector autoregression model, group lasso, thresholding

1. Introduction

We consider the problem of learning a directed network of interactions among a number of entities from time course data. A natural framework to analyze this problem uses the notion of Granger causality (Granger, 1969). Originally proposed by C.W. Granger this notion provides a statistical framework for determining whether a time series X is useful in forecasting another one Y, through a series of statistical tests. It has found wide applicability in economics, including testing relationships between money and income (Sims, 1972), government spending and taxes on economic output (Blanchard and Perotti, 2002), stock price

and volume (Hiemstra and Jones, 1994), etc. More recently the Granger causal framework has found diverse applications in biological sciences including functional genomics, systems biology and neurosciences to understand the structure of gene regulation, protein-protein interactions and brain circuitry, respectively.

It should be noted that the concept of Granger causality is based on associations between time series, and only under very stringent conditions, true causal relationships can be inferred (Pearl, 2000). Nonetheless, this framework provides a powerful tool for understanding the interactions among random variables based on time course data.

Network Granger causality (NGC) extends the notion of Granger causality among two variables to a wider class of p variables. Such extensions involving multiple time series are handled through the analysis of vector autoregressive processes (VAR) (Lütkepohl, 2005). Specifically, for p stationary time series $X_1^t, \ldots, X_p^t$, with $X^t = (X_1^t, \ldots, X_p^t)'$, one considers the class of models

$$X^t = A^1 X^{t-1} + \ldots + A^d X^{t-d} + \epsilon^t, \tag{1}$$

where $A^1, A^2, \ldots, A^d$ are $p \times p$ real-valued matrices, d is the *unknown* order of the VAR model and the innovation process satisfies $\epsilon^t \sim N(0, \sigma^2 I)$. In this model, the time series $\{X_j^t\}$ is said to be Granger causal for the time series $\{X_i^t\}$ if $A_{i,j}^h \neq 0$ for some $h = 1, \ldots, d$. Equivalently we can say that there exists an edge $X_j^{t-h} \to X_i^t$ in the underlying network model comprising of $(d+1) \times p$ nodes (see Figure 1). We call $A^1, \ldots, A^d$ the adjacency matrices from lags $1, \ldots, d$. Note that the entries A_{ij}^h of the adjacency matrices are not binary indicators of presence/absence of edges between two nodes X_i^t and X_j^{t-h}. Rather, they represent the direction and strength of influence from one node to the other.

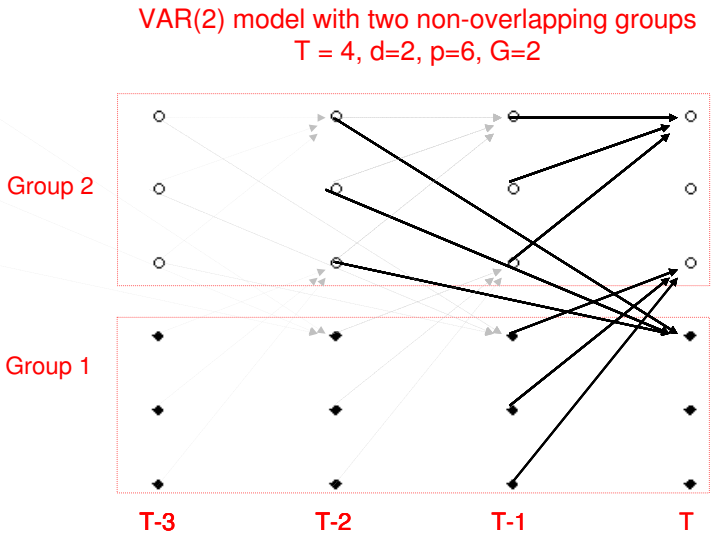


Figure 1: An example of a Network Granger causal model with two non-overlapping groups observed over T = 4 time points

The temporal structure induces a natural partial order among the nodes of this network, which in turn simplifies significantly the corresponding estimation problem (Shojaie and Michailidis, 2010a) of a directed acyclic graph. Nevertheless, one still has to deal with estimating a high-dimensional network (e.g., hundreds of genes) from a limited number of samples.

The traditional asymptotic framework of estimating VAR models requires observing a long, stationary realization $\{X^1, \ldots, X^T, T \to \infty, p, d \text{ fixed}\}$ of the p-dimensional time series. This is not appropriate in many biological applications for the following reasons. First, long stationary time series are rarely observed in these contexts. Second, the number of time series (p) being large compared to T, the task of consistent order (d) selection using standard criteria (e.g., AIC or BIC) becomes challenging. Similar issues arise in many econometric applications where empirical evidence suggests lack of stationarity over a long time horizon, although the multivariate time series exhibits locally stable distributional properties.

A more suitable framework comes from the study of panel data, where one observes several replicates of the time series, with possibly short T, across a panel of n subjects. In biological applications replicates are obtained from test subjects. In the analysis of macroeconomic variables, households or firms typically serve as replicates. After removing panel specific fixed effects one treats the replicates as independent samples, performs regression analysis under the assumption of common slope structure and studies the asymptotic properties under the regime $n \to \infty$. Recent works of Cao and Sun (2011) and Binder et al. (2005) analyze theoretical properties of short panel VARs in the low-dimensional setting ($n \to \infty$, T, p fixed).

The focus of this work is on estimating a *high-dimensional NGC model* in the panel data context (p, n large, T small to moderate). This work is motivated by two application domains, functional genomics and financial econometrics. In the first application (presented in Section 6) one is interested in reconstructing a gene regulatory network structure from time course data, a canonical problem in functional genomics (Michailidis, 2012). The second motivating example examines the composition of balance sheets of the $n = 50$ largest US banks by size, over $T = 9$ quarterly periods, which provides insight into their risk profile.

The nature of high-dimensionality in these two examples comes from both estimation of p^2 coefficients for each of the adjacency matrices $A^1, \ldots, A^d$, but also from the fact that the order of the time series d is often unknown. Thus, in practice, one must either "guess" the order of the time series (often times, it is assumed that the data is generated from a VAR(1) model, which can result in significant loss of information), or include all of the past time points, resulting in significant increase in the number of variables in cases where $d \ll T$. Thus, efficient estimation of the order of the time series becomes crucial.

Latent variable based dimension reduction techniques like principal component analysis or factor models are not very useful in this context since our goal is to reconstruct a network among the observed variables. To achieve dimension reduction we impose a group sparsity assumption on the structure of the adjacency matrices $A_1, \ldots, A_d$. In many applications, structural grouping information about the variables exists. For example, genes can be naturally grouped according to their function or chromosomal location, stocks according to their industry sectors, assets/liabilities according to their class, etc. This information can

be incorporated to the Granger causality framework through a group lasso penalty. If the group specification is correct it enables estimation of denser networks with limited sample sizes (Bach, 2008; Huang and Zhang, 2010; Lounici et al., 2011). However, the group lasso penalty can achieve model selection consistency only at a group level. In other words, if the groups are misspecified, this procedure can not perform within group variable selection (Huang et al., 2009), an important feature in many applications.

Over the past few years, several authors have adopted the framework of network Granger causality to analyze multivariate temporal data. For example, Fujita et al. (2007) and Lozano et al. (2009) employed NGC models coupled with penalized ℓ_1 regression methods to learn gene regulatory mechanisms from time course microarray data. Specifically, Lozano et al. (2009) proposed to group all the past observations, using a variant of group lasso penalty, in order to construct a relatively simple Granger network model. This penalty takes into account the average effect of the covariates over different time lags and connects Granger causality to this average effect being significant. However, it suffers from significant loss of information and makes the consistent estimation of the signs of the edges difficult (due to averaging). Shojaie and Michailidis (2010b) proposed a truncating lasso approach by introducing a truncation factor in the penalty term, which strongly penalizes the edges from a particular time lag, if it corresponds to a highly sparse adjacency matrix.

Despite recent use of NGC in applications involving high dimensional data, theoretical properties of the resulting estimators have not been fully investigated. For example, Lozano et al. (2009) and Shojaie and Michailidis (2010b) discuss asymptotic properties of the resulting estimators, but neither addresses in depth norm consistency properties, nor do they examine under what vector autoregressive structures the obtained results hold.

In this paper, we develop a general framework that accommodates different variants of group lasso penalties for NGC models. It allows for the simultaneous estimation of the order of the times series and the Granger causal effects; further, it allows for variable selection even when the groups are misspecified. In summary, the key contributions of this work are: (i) investigate in depth *sufficient conditions* that explicitly take into consideration the structure of the VAR(d) model to establish norm consistency, (ii) introduce the novel notion of *direction consistency*, which generalizes the concept of sign consistency and provides insight into the properties of group lasso estimates within a group, and (iii) use the latter notion to introduce an easy to compute thresholded variant of group lasso, that performs within group variable selection in addition to group sparsity pattern selection even when the group structure is misspecified.

All the obtained results are non-asymptotic in nature, and hence help provide insight into the properties of the estimates under different asymptotic regimes arising from varying growth rates of T, p, n, group sizes and the number of groups.

2. Model and Framework

Notation. Consider a VAR model

$$\underbrace{X^t}_{p\times 1} = \underbrace{A^1}_{p\times p} X^{t-1} + \ldots + A^d X^{t-d} + \epsilon^t, \quad \epsilon^t \sim N(0_{p\times 1}, \sigma^2 I_{p\times p}), \tag{2}$$

observed over T time points $t = 1, \ldots, T$, across n panels. The index set of the variables $\mathbb{N}_p = \{1, 2, \ldots, p\}$ can be partitioned into G non-overlapping groups $\mathcal{G}_g$, i.e., $\mathbb{N}_p = \cup_{g=1}^{G} \mathcal{G}_g$ and $\mathcal{G}_g \cap \mathcal{G}_{g'} = \phi$ if $g \neq g'$. Also $k_g = |\mathcal{G}_g|$ denotes the size of the g^{th} group with $k_{max} = \max_{1 \leq g \leq G} k_g$. In general, we use $\lambda_{\min}$ and $\lambda_{\max}$ to denote the minimum and maximum of a finite collection of numbers $\lambda_1, \ldots, \lambda_m$.

For any matrix A, we denote the i^{th} row by $A_{i:}$, j^{th} column by $A_{:j}$ and the collection of rows (columns) corresponding to the g^{th} group by $A_{[g]:}$ ($A_{:[g]}$). The transpose of a matrix A is denoted by A' and its Frobenius norm by $||A||_F$. For a symmetric/Hermitian matrix Σ, its maximum and minimum eigenvalues are denoted by $\Lambda_{\min}(\Sigma)$ and $\Lambda_{\max}(\Sigma)$, respectively. The symbol $A^{1:h}$ is used to denote the concatenated matrix $\left[A^1 : \cdots : A^h\right]$, for any $h > 0$. For any matrix or vector D, $\|D\|_0$ denotes the number of non-zero coordinates in D. For notational convenience, we reserve the symbol $\|.\|$ to denote the ℓ_2 norm of a vector and/or the spectral norm of a matrix. For a pre-defined set of non-overlapping groups $\mathcal{G}_1, \ldots, \mathcal{G}_G$ on $\{1, \ldots, p\}$, the mixed norms of vectors $v \in \mathbb{R}^p$ are defined as $\|v\|_{2,1} = \sum_{g=1}^{G} \|v_{[g]}\|$ and $\|v\|_{2,\infty} = \max_{1 \leq g \leq G} \|v_{[g]}\|$. Also for any vector β, we use β_j to denote its j^{th} coordinate and $\beta_{[g]}$ to denote the coordinates corresponding to the g^{th} group. We also use $supp(v)$ to denote the support of v, i.e., $supp(v) = \{j \in \{1, \ldots, p\} | v_j \neq 0\}$.

Network Granger causal (NGC) estimates with group sparsity. Consider n replicates from the NGC model (2), and denote the $n \times p$ observation matrix at time t by $\mathcal{X}^t$. In econometric applications the data on p economic variables across n panels (firms, households etc.) can be observed over T time points. For time course microarray data one typically observes the expression levels of p genes across n subjects over T time points. After removing the panel specific fixed effects one assumes the common slope structure and independence across the panels. The data are high-dimensional if either T or p is large compared to n. In such a scenario, we assume the existence of an underlying group sparse structure, i.e., for every $i = 1, \ldots, p$, the support of the i^{th} row of $A^{1:T-1} = \left[A^1 : \cdots : A^{T-1}\right]$ in the model (2) can be covered by a small number of groups s_i, where $s_i \ll (T-1)G$. Note that the groups can be misspecified in the sense that the coordinates of a group covering the support need not be all non-zero. Hence, for a properly specified group structure we shall expect $s_i \ll \|A^{1:T}_{i:}\|_0$. On the contrary, with many misspecified groups, s_i can be of the same order, or even larger than $\|A^{1:T}_{i:}\|_0$.

Learning the network of Granger causal effects $\{(i,j) \in \{1, \ldots, p\} : A^t_{ij} \neq 0 \text{ for some } t\}$ is equivalent to recovering the correct sparsity pattern in $A^{1:(T-1)}$ and consistently estimating the non-zero effects A^t_{ij}. In the high-dimensional regression problems this is achieved by simultaneous regularization and selection operators like lasso and group lasso. The group Granger causal estimates of the adjacency matrices $A^1, \ldots, A^{T-1}$ are obtained by solving the following optimization problem

$$\hat{A}^{1:T-1} = \underset{A^1, \cdots, A^{T-1}}{\operatorname{argmin}} \frac{1}{2n} \left\| \mathcal{X}^T - \sum_{t=1}^{T-1} \mathcal{X}^{T-t} \left(A^t\right)' \right\|_F^2 + \lambda \sum_{t=1}^{T-1} \sum_{i=1}^{p} \sum_{g=1}^{G} w^t_{i,g} \|A^t_{i:[g]}\|, \qquad (3)$$

where $\mathcal{X}^t$ is the $n \times p$ observation matrix at time t, constructed by stacking n replicates from the model (2), w^t is a $p \times G$ matrix of suitably chosen weights and λ is a common

regularization parameter. The optimization problem can be separated into the following p penalized regression problems:

$$\hat{A}_{i:}^{1:T-1} = \underset{\theta^1,\cdots,\theta^{T-1}\in\mathbb{R}^p}{\operatorname{argmin}} \frac{1}{2n}\|\mathcal{X}_{:i}^T - \sum_{t=1}^{T-1}\mathcal{X}^{T-t}\theta^t\|^2 + \lambda\sum_{t=1}^{T-1}\sum_{g=1}^{G} w_{i,g}^t\|\theta_{[g]}^t\|, \quad i=1,\cdots,p. \tag{4}$$

The order d of the VAR model is estimated as $\hat{d} = \max_{1\le t\le T-1}\{t : \hat{A}^t \neq \mathbf{0}\}$.

Different choices of weights $w_{i:g}^t$ lead to different variants of NGC estimates. The regular NGC estimates correspond to the choices $w_{i,g}^t = 1$ or $\sqrt{k_g}$, while for adaptive group NGC estimates the weights are chosen as $w_{i,g}^t = \left\|\hat{A}_{i:[g]}^t\right\|^{-1}$, where $\hat{A}^t$ are obtained from a regular NGC estimation. For $\hat{A}_{i:[g]}^t = \mathbf{0}$, the weight $w_{i,g}^t$ is infinite, which is interpreted as discarding the variables in group g from the optimization problem.

Thresholded NGC estimates are calculated by a two-stage procedure. The first stage involves a regular NGC estimation procedure. The second stage uses a bi-level thresholding strategy on the estimates $\hat{A}^t$. First, the estimated groups with ℓ_2 norm less than a threshold ($\delta_{grp} = c\lambda$, $c > 0$) are set to zero. The second level of thresholding (within group) is applied if the *a priori* available grouping information is not entirely reliable. $\hat{A}_{ij}^t$within an estimated group $\hat{A}_{i:[g]}^t$ is thresholded to zero if $\left|\hat{A}_{ij}^t\right| / \left\|\hat{A}_{i:[g]}^t\right\|$ is less than a threshold $\delta_{misspec} \in (0,1)$. So, for every $t = 1,\ldots,T-1$, if $j \in \mathcal{G}_g$, the thresholded NGC estimates are

$$\tilde{A}_{ij}^t = \hat{A}_{ij}^t I\left\{\left|\hat{A}_{ij}^t\right| \ge \delta_{misspec}\left\|\hat{A}_{i:[g]}^t\right\|\right\} I\left\{\left\|\hat{A}_{i:[g]}^t\right\| \ge \delta_{grp}\right\}.$$

The tuning parameters λ_{grp} and $\delta_{misspec}$ are chosen via cross-validation. The rationale behind this thresholding strategy is discussed in Section 4.

3. Estimation Consistency of NGC estimates

In this section we establish the norm consistency of regular group NGC estimates. The regular NGC estimates in (3) are obtained by solving p separate group lasso programs with a *common* design matrix $\mathbf{X}_{n\times p(T-1)} = [\mathcal{X}^1 : \cdots : \mathcal{X}^{T-1}]$. This design matrix has $\bar{p} = (T-1)p$ columns which can be partitioned into $\bar{G} = (T-1)G$ groups $\{\mathcal{G}_1,\ldots,\mathcal{G}_{\bar{G}}\}$. We denote the sample Gram matrix by $C = \mathcal{X}'\mathcal{X}/n$. For the i^{th} optimization problem, these $\bar{G} = (T-1)G$ groups are penalized by $\lambda_{(t-1)G+g} := \lambda\, w_{i,g}^t$, $1 \le t \le T-1$, $1 \le g \le G$, with the choice of weights $w_{i,g}^t$ described in Section 2. Following Lounici et al. (2011) one can establish a non-asymptotic upper bound on the ℓ_2 estimation error of the NGC estimates $\hat{A}^t$ under certain restricted eigenvalue (RE) assumptions. These assumptions are common in the literature of high-dimensional regression (Lounici et al., 2011; Bickel et al., 2009; van de Geer and Bühlmann, 2009) and are known to be sufficient to guarantee consistent estimation of the regression coefficients even when the design matrix is singular. Of main interest, however, is to investigate the validity of these assumptions in the context of NGC models. This issue is addressed in Proposition 3.2.

For $L > 0$, we say that a **Restricted Eigenvalue** (RE) assumption RE(s, L) is satisfied if there exists a positive number $\phi_{RE} = \phi_{RE}(s) > 0$ such that

$$\min_{\substack{J \subset \mathbb{N}_{\bar{G}}, |J| \le s \\ \Delta \in \mathbb{R}^{\bar{p}} \setminus \{\mathbf{0}\}}} \left\{ \frac{\|\mathbf{X}\Delta\|}{\sqrt{n}\|\Delta_{[J]}\|} : \sum_{g \in J^c} \lambda_g \|\Delta_{[g]}\| \le L \sum_{g \in J} \lambda_g \|\Delta_{[g]}\| \right\} \ge \phi_{RE}. \tag{5}$$

The following proposition provides a non-asymptotic upper bound on the ℓ_2-estimation error of the group NGC estimates under RE assumptions. The proof follows along the lines of Lounici et al. (2011) and is delegated to Appendix C.

Proposition 3.1 *Consider a regular NGC estimation problem* (4) *with* $s_{\max} = \max_{1 \le i \le p} s_i$ *and* $s = \sum_{i=1}^{p} s_i$. *Suppose* λ *in* (3) *is chosen large enough so that for some* $\alpha > 0$,

$$\lambda_g \ge \frac{2\sigma}{\sqrt{n}} \sqrt{\left\|C_{[g][g]}\right\|} \left(\sqrt{k_g} + \frac{\pi}{\sqrt{2}} \sqrt{\alpha \log \bar{G}} \right) \quad \textit{for every } g \in \mathbb{N}_{\bar{G}}, \tag{6}$$

Also assume that the common design matrix $\mathbf{X} = [\mathcal{X}^1 : \cdots : \mathcal{X}^{T-1}]$ *in the* p *regression problems* (4) *satisfy* $RE(2s_{\max}, 3)$. *Then, with probability at least* $1 - 2p\bar{G}^{1-\alpha}$,

$$\left\| \hat{A}^{1:T-1} - A^{1:T-1} \right\|_F \le \frac{4\sqrt{10}}{\phi_{RE}^2(2s_{\max})} \frac{\lambda_{\max}^2}{\lambda_{\min}} \sqrt{s}. \tag{7}$$

Remark. Consider a high-dimensional asymptotic regime where $\bar{G} \asymp n^B$ for some $B > 0$, $k_{\max}/k_{\min} = O(1)$, $s = O(n^{a_1})$ and $k_{\max} = O(n^{a_2})$ with $0 < a_1,\ a_2 < a_1 + a_2 < 1$ so that the total number of non-zero effects is $o(n)$. If $\{\|C_{[g][g]}\|,\ g \in \mathbb{N}_{\bar{G}}\}$ are bounded above (often accomplished by standardizing the data) and $\phi_{RE}^2(2s_{\max})$ is bounded away from zero (see Proposition 3.2 for more details), then the NGC estimates are norm consistent for any choice of $\alpha > 2 + a_2/B$.

Note that group lasso achieves faster convergence rate (in terms of estimation and prediction error) than lasso if the groups are appropriately specified. For example, if all the groups are of equal size k and $\lambda_g = \lambda$ for all g, then group lasso can achieve an ℓ_2 estimation error of order $O\left(\sqrt{s}(\sqrt{k} + \sqrt{\log \bar{G}})/\sqrt{n}\right)$. In contrast, lasso's error is known to be of the order $O\left(\sqrt{\|A^{1:d}\|_0 \log \bar{p}/n}\right)$, which establishes that group lasso has a lower error bound if $s \ll \|A^{1:d}\|_0$. On the other hand, lasso will have a lower error bound if $s \asymp \|A^{1:d}\|_0$, i.e., if the groups are highly misspecified.

Validity of RE assumption in Group NGC problems. In view of Theorem 3.1, it is important to understand how stringent the RE condition is in the context of NGC problems. It is also important to find a lower bound on the RE coefficient ϕ_{RE}, as it affects the convergence rate of the NGC estimates. For the panel-VAR setting, we can rigorously establish that the RE condition holds with overwhelming probability, as long as $n,\ p$ grow at the same rate required for ℓ_2-consistency.

The following proposition achieves this objective in two steps. Note that each row of the design matrix $\mathbf{X}$ (common across the p regressions) is independently distributed as $N(\mathbf{0}, \Sigma)$ where Σ is the variance-covariance matrix of the $(T-1)p$-dimensional random variable

$\left((X^1)', \ldots, (X^{T-1})'\right)'$. First, we exploit the spectral representation of the stationary VAR process to provide a lower bound on the minimum eigenvalue of Σ. In the next step, we establish a suitable deviation bound on $\mathbf{X} - \Sigma$ to prove that $\mathbf{X}$ satisfies RE condition with high probability for sufficiently large n.

Proposition 3.2 *(a) Suppose the VAR(d) model of* (2) *is stable, stationary. Let* Σ *be the variance-covariance matrix of the* $(T-1)p$*-dimensional random variable* $\left((X^1)', \ldots, (X^{T-1})'\right)'$. *Then the minimum eigenvalue of* Σ *satisfies*

$$\Lambda_{min}(\Sigma) \geq \sigma^2 \left[\max_{\theta \in [-\pi, \pi]} \|\mathcal{A}(e^{-i\theta})\|\right]^{-2} \geq \sigma^2 \left[1 + \sum_{t=1}^{d} \|A^t\|\right]^{-2} \geq \sigma^2 \left[1 + \frac{1}{2}(\mathbf{v}_{in} + \mathbf{v}_{out})\right]^{-2},$$

where $\mathcal{A}(z) := I - A^1 z - A^2 z^2 - \ldots - A^d z^d$ *is the reverse characteristic polynomial of the VAR(d) process, and* $\mathbf{v}_{in}$, $\mathbf{v}_{out}$ *are the maximum incoming and outgoing effects at a node, cumulated across different lags*

$$\mathbf{v}_{in} = \sum_{t=1}^{d} \max_{1 \leq i \leq p} \sum_{j=1}^{p} |A^t_{ij}|, \qquad \mathbf{v}_{out} = \sum_{t=1}^{d} \max_{1 \leq j \leq p} \sum_{i=1}^{p} |A^t_{ij}|.$$

(b) In addition, suppose the replicates from different panels are i.i.d. Then, for any $s > 0$, *there exist universal positive constants* c_i *such that if the sample size* n *satisfies*

$$n \quad > \frac{\Lambda^2_{\max}(\Sigma)}{\Lambda^2_{\min}(\Sigma)} (2 + L\lambda_{\max}/\lambda_{\min})^4 \, c_0 s(k_{\max} + c_1 \log(e\bar{G}/2s)),$$

then $\mathbf{X}$ *satisfies* $RE(s, L)$ *with* $\phi^2_{RE} \geq \Lambda_{\min}(\Sigma)/2$ *with probability at least* $1 - c_2 \exp(-c_3 n)$.

Remark. Proposition 3.2 has two interesting consequences. First, it provides a lower bound on the RE constant ϕ_{RE} which is independent of T. So if the high dimensionality in the Granger causal network arises only from the time domain and not the cross-section ($T \to \infty$, p, G fixed), the stationarity of the VAR process guarantees that the rate of convergence depends only on the true order (d), and not T. Second, this result shows that the NGC estimates are consistent even if the node capacities $\mathbf{v}_{in}$ and $\mathbf{v}_{out}$ grow with n, p at an appropriate rate.

4. Variable Selection Consistency of NGC estimates

In view of (4), to study the variable selection properties of NGC estimates it suffices to analyze the variable selection properties of p generic group lasso estimates with a common design matrix.

The problem of group sparsity selection has been thoroughly investigated in the literature (Wei and Huang, 2010; Lounici et al., 2011). The issue of selection and sign consistency within a group, however, is still unclear. Since group lasso does not impose sparsity within a group, all the group members are selected together (Huang et al., 2009) and it is not clear which ones are recovered with correct signs. This also leads to inconsistent variable selection if a group is misspecified, i.e., not all the members within a group have non-zero

effect. Several alternate penalized regression procedures have been proposed to overcome this shortcoming (Breheny and Huang, 2009; Huang et al., 2009). The main idea behind these procedures is to combine ℓ_2 and ℓ_1 norms in the penalty to encourage sparsity at both group and variable level. These estimators involve nonconvex optimization problems and are computationally expensive. Also their theoretical properties in a high dimensional regime are not well studied.

We take a different approach to deal with the issue of group misspecification. Although the group lasso penalty does not perform exact variable selection within groups, it performs regularization and shrinks the individual coefficients. We utilize this regularization to detect misspecification within a group. To this end, we formulate a generalized notion of sign consistency, henceforth referred as "direction consistency", that provides insight into the properties of group lasso estimates within a single group. Subsequently, these properties are used to develop a simple, easy to compute, thresholded variant of group lasso which, in addition to group selection, achieves variable selection and sign consistency within groups.

We consider a generic group lasso regression problem of the linear model $y = X\beta^0 + \epsilon$ with p variables partitioned into G non-overlapping groups $\{\mathcal{G}_1, \ldots, \mathcal{G}_G\}$ of size k_g, $g = 1, \ldots, G$. Without loss of generality, we assume $\beta^0_{[g]} \neq \mathbf{0}$ for $g \in S = \{1, 2, \ldots, s\}$ and $\beta^0_{[g]} = \mathbf{0}$ for all $g \notin S$ and consider the following group lasso estimate of β^0:

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \frac{1}{2n} \|\mathbf{Y} - \mathbf{X}\beta\|^2 + \sum_{g=1}^{G} \lambda_g \|\beta_{[g]}\|, \tag{8}$$

$$\underbrace{\beta^0}_{p\times 1} = [\underbrace{\beta^0_{[1]}, \ldots, \beta^0_{[s]}}_{k_1+\ldots+k_s=q}, \underbrace{\mathbf{0}, \ldots, \mathbf{0}}_{p-q}] = [\beta^0_{(1)} : \beta^0_{(2)}], \tag{9}$$

$$\underbrace{\mathbf{X}}_{n\times p} = [\underbrace{\mathbf{X}_{(1)}}_{n\times q} : \underbrace{\mathbf{X}_{(2)}}_{n\times(p-q)}], \qquad C = \frac{1}{n}\mathbf{X}'\mathbf{X} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}. \tag{10}$$

Direction Consistency. For an m-dimensional vector $\tau \in \mathbb{R}^m \backslash \{\mathbf{0}\}$ define its direction vector $D(\tau) = \tau/\|\tau\|$, $D(\mathbf{0}) = \mathbf{0}$. In the context of a generic group lasso regression (10), for a group $g \in S$ of size k_g, $D(\beta^0_{[g]})$ indicates the direction of influence of $\beta^0_{[g]}$ at a group level in the sense that it reflects the relative importance of the influential members within the group. Note that for $k_g = 1$ the function $D(\cdot)$ simplifies to the usual $sgn(\cdot)$ function.

Definition. An estimate $\hat{\beta}$ of a generic group lasso problem (8) is ***direction consistent*** at a rate δ_n, if there exists a sequence of positive real numbers $\delta_n \to 0$ such that

$$\mathbb{P}\left(\|D(\hat{\beta}_{[g]}) - D(\beta^0_{[g]})\| < \delta_n,\ \forall g \in S,\ \hat{\beta}_{[g]} = \mathbf{0},\ \forall g \notin S\right) \to 1 \text{ as } n, p \to \infty. \tag{11}$$

Now suppose $\hat{\beta}$ is a direction consistent estimator. Consider the set $\tilde{S}^n_g := \{j \in \mathcal{G}_g : |{\beta^0}_j| \,/\, \|{\beta^0}_{[g]}\| > \delta_n\}$. $\tilde{S}^n_g$ can be viewed as a collection of influential group members within a group $\mathcal{G}_g$, which are "detectable" with a sample of size n. Then, it readily follows from the definition that

$$\mathbb{P}(sgn(\hat{\beta}_j) = sgn(\beta_j),\ \forall j \in \tilde{S}^n_g, \forall g \in \{1, \ldots, s\}) \to 1 \text{ as } n, p \to \infty. \tag{12}$$

The latter observation connects the precision of group lasso estimates to the accuracy of *a priori* available grouping information. In particular, if the pre-specified grouping structure is correct, i.e., all the members within a group have non-zero effects, then for a sufficiently large sample size we have $\tilde{S}_g^n = \mathcal{G}_g$ for all $g \in S$. Hence, if the group lasso estimate is direction consistent, it will correctly estimate the sign of all the variables in the support. On the other hand, in case of a misspecified *a priori* grouping structure (numerous zero coordinates in β_g for $g \in S$), group lasso will correctly estimate only the signs of the influential group members. This argument on zero vs. non-zero effects can be generalized to strong vs. weak effects, as well.

Example. We demonstrate the property of direction consistency using a small example. Consider a linear model with 8 predictors

$$y = 0.5x_1 - 3x_2 + 3x_3 + x_4 - 2x_5 + 3x_8 + e, \quad e \sim N(0,1).$$

The coefficient vector β^0 is partitioned into four groups of size 2, viz., $(0.5, -3)$, $(3, 1)$, $(-2, 0)$ and $(0, 3)$. The last two groups are misspecified. We generated $n = 25$ samples from this model and ran group lasso regression with the above group structure. Figure 2 shows the true coefficient vectors (solid) and their estimates (dashed) from five iterations of the above exercise. Note that even though the ℓ_2 errors between $\beta^0_{[g]}$ and $\hat{\beta}_{[g]}$ vary largely across the four groups, the distance between their projections on the unit circle, $\left\| D(\beta^0_{[g]}) - D(\hat{\beta}_{[g]}) \right\|$, are comparatively stable across groups. In fact, Theorem 4.1 shows that under certain irrepresentable conditions (IC) on the design matrix, it is possible to find a uniform (over all $g \in S$) upper bound δ_n on the ℓ_2 gap of these direction vectors. This motivates a natural thresholding strategy to correct for the misspecification in groups (cf. Proposition 4.2). Even though a group $\beta^0_{[g]}$ is misspecified (i.e., lies on a coordinate axis), direction consistency ensures, with high probability, that the corresponding coordinate in $D(\hat{\beta}_{[g]})$ will be smaller than a threshold δ_n which is common across all groups in the support.

Group Irrepresentable Conditions (IC). Next, we define the IC required for direction consistency of group lasso estimates. Irrepresentable conditions are common in the literature of high-dimensional regression problems (Zhao and Yu, 2006; van de Geer and Bühlmann, 2009) and are shown to be sufficient (and essentially necessary) for selection consistency of the lasso estimates. Further these conditions are known to be satisfied with high probability, if the population analogue of the Gram matrix belongs to the Toeplitz family (Zhao and Yu, 2006; Wainwright, 2009). In NGC estimation the population analogue of the Gram matrix $\Sigma = Var(\mathbf{X}^{1:(T-1)})$ is block Toeplitz, so the irrepresentable assumptions are natural candidates for studying selection consistency of the estimates. Consider the notations of (8) and (10). Define $K = diag\,(\lambda_1 \mathbf{I}_{k_1}, \lambda_2 \mathbf{I}_{k_2}, \ldots, \lambda_s \mathbf{I}_{k_s})$.

Uniform Irrepresentable Condition (IC) is satisfied if there exists $0 < \eta < 1$ such that for all $\tau \in \mathbb{R}^q$ with $\|\tau\|_{2,\infty} = \max_{1 \le g \le s} \|\tau_{[g]}\|_2 \le 1$,

$$\frac{1}{\lambda_g} \left\| \left[C_{21} (C_{11})^{-1} K \tau \right]_{[g]} \right\| < 1 - \eta, \ \forall g \notin S = \{1, \ldots, s\}. \tag{13}$$

Note that the definition reverts to the usual IC for lasso when all groups correspond are singletons.

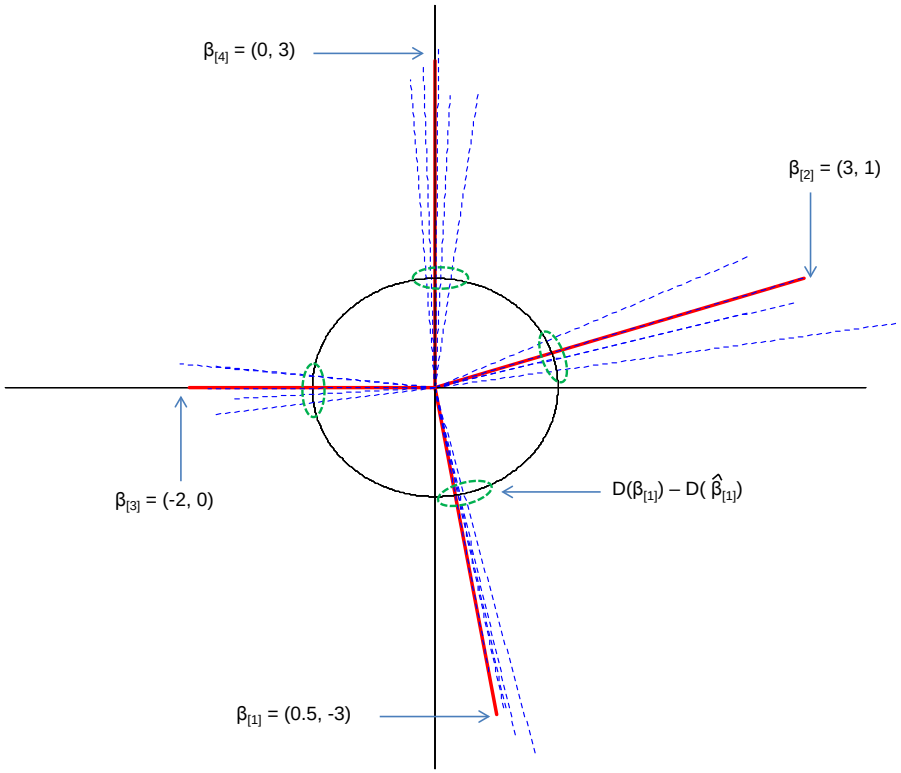


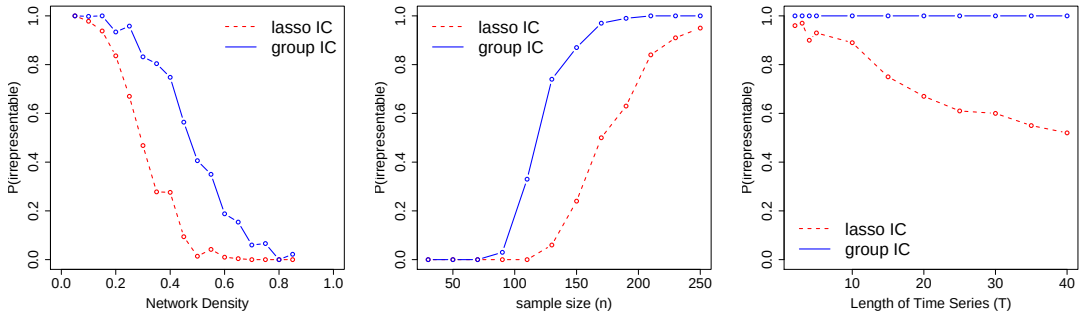
Figure 2: Example demonstrating direction consistency

The IC is more stringent than the RE condition and is rarely met if the underlying model is not sparse. It can be shown that a slightly weaker version of this condition is necessary for direction consistency. We refer the readers to Appendix D for further discussion on the different irrepresentable assumptions and their properties. Numerical evidence suggests that the group IC tends to be less stringent than the IC required for the selection consistency of lasso. We illustrate this using three small simulated examples.

Simulation 1. We constructed group sparse NGC models with $T = 5$, $p = 21$, $G = 7, k_g = 3$ and different levels of network densities, where the network edges were selected at random and scaled so that $\|A^1\| = 0.1$. For each of these models, we generated 100 samples of size $n = 150$ and calculated the proportions of times the two types of irrepresentable conditions were met. The results are displayed in Figure 3a.

Simulation 2. We selected a VAR(1) model from the above class and generated samples of size $n = 20, 50, \ldots, 250$. Figure 3b displays the proportions of times (based on 100 simulations) the two ICs were met.

Simulation 3. We generated $n = 200$ samples from the VAR(1) model of example 2 for $T = 2, 3, 4, 5, 10, \ldots, 40$. Figure 3c displays the proportions of times (based on 100 simulations) the two ICs were met.



(a) IC and Network Density (b) IC and Sample Size (c) IC and Time Series length

Figure 3: Comparison of lasso and group irrepresentable conditions in the context of group sparse NGC models. (a) group ICs tend to be met for dense networks where lasso IC fails to meet. (b) For the same network group IC is met with smaller sample size than required by lasso. (c) For longer time series group IC is satisfied more often than lasso IC.

Selection consistency for generic group lasso estimates. For simplicity, we discuss the selection consistency properties of a generic group lasso regression problem with a common tuning parameter across groups, i.e., $\lambda_g = \lambda$ for every $g \in \mathbb{N}_G$. Similar results can be obtained for more general choices of the tuning parameters.

Theorem 4.1 *Assume that the group uniform IC holds with $1-\eta$ for some $\eta > 0$. Then, for any choice of $\alpha > 0$,*

$$\lambda \geq \max_{g \notin S} \frac{1}{\eta} \frac{\sigma}{\sqrt{n}} \sqrt{\left\| (C_{22})_{[g][g]} \right\|} \left(\sqrt{k_g} + \frac{\pi}{\sqrt{2}} \sqrt{\alpha \log G} \right) \quad \text{and}$$

$$\delta_n \geq \max_{g \in S} \frac{1}{\left\| \beta^0_{[g]} \right\|} \left(\lambda \sqrt{s} \left\| (C_{11})^{-1} \right\| + \frac{\sigma}{\sqrt{n}} \sqrt{\left\| (C_{11})^{-1}_{[g][g]} \right\|} \left(\sqrt{k_g} + \frac{\pi}{\sqrt{2}} \sqrt{\alpha \log G} \right) \right),$$

with probability greater than $1 - 4G^{1-\alpha}$, there exists a solution $\hat{\beta}$ satisfying

1. *$\hat{\beta}_{[g]} = 0$ for all $g \notin S$,*
2. *$\left\| \hat{\beta}_{[g]} - \beta^0_{[g]} \right\| < \delta_n \left\| \beta^0_{[g]} \right\|$, and hence $\left\| D(\hat{\beta}_{[g]}) - D(\beta^0_{[g]}) \right\| < 2\delta_n$, for all $g \in S$. If $\delta_n < 1$, then $\hat{\beta}_{[g]} \neq 0$ for all $g \in S$.*

Remark. The tuning parameter λ can be chosen of the same order as required for ℓ_2 consistency to achieve selection consistency within groups in the sense of (12). Further, with the above choice of λ, δ_n can be chosen of the order of $O(\sqrt{s}(\sqrt{k_{max}} + \sqrt{\log G})/\sqrt{n})$. Thus, group lasso correctly identifies the group sparsity pattern and is direction consistent if $\sqrt{s}(\sqrt{k_{max}} + \sqrt{\log G})/\sqrt{n} \to 0$, the same scaling required for ℓ_2 consistency.

Thresholding in Group NGC estimators. As described in Section 2, regular group NGC estimates can be thresholded both at the group and coordinate levels. The first level of thresholding is motivated by the fact that lasso can select too many false positives [cf. van de Geer et al. (2011), Zhou (2010) and the references therein]. The second level of thresholding employs the direction consistency of regular group NGC estimates to perform within group variable selection with high probability. The following proposition demonstrates the benefit of these two types of thresholding. The second result is an immediate corollary of Theorem 4.1. Proof of the first result (thresholding at group level) requires some additional notations and is delegated to Appendix E.

Theorem 4.2 *Consider a generic group lasso regression problem* (8) *with common tuning parameter* $\lambda_g = \lambda$.
(i) Assume the RE(s, 3) condition of (5) *holds with a constant* ϕ_{RE} *and define* $\hat{\beta}_{[g]}^{thgrp} = \hat{\beta}_{[g]}\mathbf{1}_{\|\hat{\beta}_{[g]}\|>4\lambda}$. *If* $\hat{S} = \{g \in \mathbb{N}_G : \hat{\beta}_{[g]}^{thgrp} \neq \mathbf{0}\}$, *then* $|\hat{S}\backslash S| \leq \frac{s}{\phi_{RE}^2/12}$, *with probability at least* $1 - 2G^{1-\alpha}$.
(ii) Assume that uniform IC holds with $1 - \eta$ *for some* $\eta > 0$. *Choose* λ *and* δ_n *as in Theorem 4.1 and define*

$$\hat{\beta}_j^{thgrp} = \hat{\beta}_j \mathbf{1}\{|\hat{\beta}_j|/\|\hat{\beta}_{[g]}\| > 2\,\delta_n\} \text{ for all } j \in \mathcal{G}_g.$$

Then $sgn(\beta_j^0) = sgn(\hat{\beta}_j^{thgrp})\ \forall\, j \in \mathbb{N}_p$ *with probability at least* $1 - 4G^{1-\alpha}$, *if* $\min_{j \in supp(\beta^0)} |\beta_j^0| > 2\delta_n \|\beta_{[g]}^0\|$ *for all* $j \in \mathcal{G}_g$, *i.e., if the effect of every non-zero member in a group is "visible" relative to the total effect from the group.*

5. Performance Evaluation

We evaluate the performances of regular, adaptive and thresholded variants of the group NGC estimators through an extensive simulation study, and compare the results to those obtained from lasso estimates. The R package `grpreg` (Breheny and Huang, 2009) was used to obtain the group lasso estimates. The settings considered are:
(a) *Balanced groups of equal size*: i.i.d samples of size $n = 60, 110, 160$ are generated from lag-2 ($d = 2$) VAR models on $T = 5$ time points, comprising of $p = 60, 120, 200$ nodes partitioned into groups of equal size in the range 3-5.
(b) *Unbalanced groups:* We retain the same setting as before, however the corresponding node set is partitioned into one larger group of size 10 and many groups of size 5.
(c) *Misspecified balanced groups:* i.i.d samples of size $n = 60, 110, 160$ are generated from lag-2 ($d = 2$) VAR models on $T = 10$ time points, comprising of $p = 60, 120$ nodes partitioned into groups of size 6. Further, for each group there is a 30% misspecification rate, namely that for every parent group of a downstream node, 30% of the group members do not exert any effect on it.

Using a 19 : 1 sample-splitting, the tuning parameter λ is chosen from an interval of the form $[C_1\lambda_e, C_2\lambda_e]$, $C_1, C_2 > 0$, where $\lambda_e = \sqrt{2 \log p/n}$ for lasso and $\sqrt{2 \log G/n}$ for group lasso. The thresholding parameters are selected as $\delta_{grp} = 0.7\lambda\sigma$ at the group level and $\delta_{misspec} = n^{-0.2}$ within groups. These parameters are chosen by conducting a 20-fold cross-validation on independent tuning data sets of same sizes, using intervals of the form

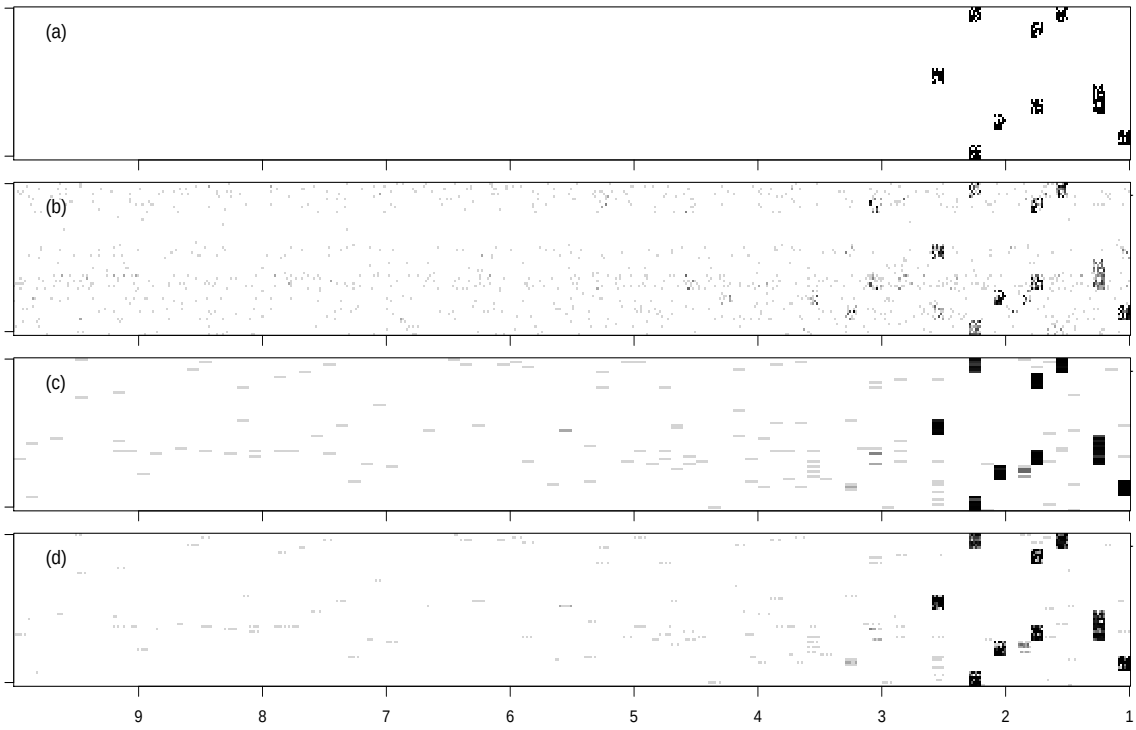


Figure 4: Estimated adjacency matrices of a misspecified NGC model with p = 60, T = 10, n = 60: (a) True, (b) Lasso, (c) Group Lasso, (d) Thresholded Group Lasso. The grayscale represents the proportion of times an edge was detected in 100 simulations.

$[C_3\lambda, C_4\lambda]$ for δ_{grp} and $\{n^{-\delta}, \delta \in [0,1]\}$ for $\delta_{misspec}$. Finally, within group thresholding is applied only when the group structure is misspecified.

The following performance metrics were used for comparison purposes: (i) $Precision = TP/(TP+FP)$, (ii) $Recall = TP/(TP+FN)$ and (iii) Matthew's Correlation coefficient (MCC) defined as

$$\frac{(TP \times TN) - (FP \times FN)}{((TP+FP) \times (TP+FN) \times (TN+FP) \times (TN+FN))^{1/2}},$$

where TP, TN, FP and FN correspond to true positives, true negatives, false positives and false negatives in the estimated network, respectively. The average and standard deviations (over 100 replicates) of the performance metrics are presented for each setup.

The results for the balanced settings are given in Table 1. The Recall for $p = 60$ shows that even for a network with $60 \times (5-1) = 240$ nodes and $|E| = 351$ true edges, the group NGC estimators recover about 71% of the true edges with a sample size as low as $n = 60$, while lasso based NGC estimates recover only 31% of the true edges. The three group NGC estimates have comparable performances in all the cases. However thresholded lasso shows slightly higher precision than the other group NGC variants for smaller sample sizes (e.g., $n = 60, p = 200$). The results for $p = 60, n = 110$ also display that lower precision of

		$p = 60$, $\|E\| = 351$ Group Size=3			$p = 120$, $\|E\| = 1404$ Group Size=3			$p = 200$, $\|E\| = 3900$ Group Size=5		
	n	160	110	60	160	110	60	160	110	60
P	Lasso	80(2)	75(2)	66(4)	69(1)	62(2)	52(2)	52(1)	47(1)	38(1)
	Grp	95(2)	91(4)	83(7)	91(3)	80(5)	68(7)	78(4)	72(3)	59(6)
	Thgrp	96(1)	92(3)	86(6)	93(3)	83(5)	70(7)	82(4)	76(3)	64(6)
	Agrp	96(2)	92(4)	83(7)	92(3)	82(5)	69(7)	81(3)	74(3)	60(6)
R	Lasso	71(2)	54(2)	31(2)	54(1)	40(1)	22(1)	38(1)	28(1)	15(1)
	Grp	99(1)	93(3)	71(7)	91(2)	81(2)	48(8)	84(1)	70(2)	41(4)
	Thgrp	99(1)	93(3)	71(7)	91(2)	81(2)	48(8)	84(2)	69(2)	41(3)
	Agrp	99(1)	93(3)	71(7)	91(2)	81(2)	47(8)	84(1)	69(2)	40(4)
MCC	Lasso	75(2)	63(2)	45(3)	60(1)	49(1)	33(1)	43(1)	35(1)	23(1)
	Grp	97(1)	92(3)	76(5)	91(1)	80(2)	56(2)	81(2)	70(2)	48(2)
	Thgrp	98(1)	93(2)	78(5)	92(1)	81(2)	57(3)	83(2)	72(2)	50(3)
	Agrp	97(1)	92(3)	76(5)	91(1)	81(2)	56(3)	82(2)	71(2)	48(2)
ERR	Lasso	10.5	11.3	13.9	16.63	17.37	16.69	19.79	20	18.52
LAG	Grp	3.19	6.95	12.76	4.86	10.77	12.65	4.21	5.27	7.8
	Thgrp	2.83	5.87	10.01	3.98	9.03	11.19	3.06	3.91	5.68
	Agrp	3.13	6.89	12.59	4.63	10.37	12.34	3.58	4.87	7.59

Table 1: Performance of different regularization methods in estimating graphical Granger causality with **balanced** group sizes and no misspecification; $d = 2$, $T = 5$, $SNR = 1.8$. Precision (P), Recall (R), MCC are given in percentages (numbers in parentheses give standard deviations). ERR LAG gives the error associated with incorrect estimation of VAR order.

lasso is caused partially by its inability to estimate the order of the VAR model correctly, as measured by ERR LAG=Number of falsely connected edges from lags beyond the true order of the VAR model divided by the number of edges in the network ($|E|$). This finding is nicely illustrated in Figure 4 and Table 1. The group penalty encourages edges from the nodes of the same group to be picked up together. Since the nodes of the same group are also from the same time lag, the group variants have substantially lower ERR LAG. For example, average ERR LAG of lasso for $p = 200,\ n = 160$ is 19.79% while the average ERR LAGs for the group lasso variants are in the range $3.06\% - 4.21\%$.

The results for the unbalanced networks are given in Table 2. As in the balanced group setup, in almost all the simulation settings the group NGC variants outperform the lasso estimates with respect to all three performance metrics. However the performances of the different variants of group NGC are comparable and tend to have higher standard deviations than the lasso estimates. Also the average ERR LAGs for the group NGC variants are substantially lower than the average ERR LAG for lasso demonstrating the advantage of group penalty. Although the conclusions regarding the comparisons of lasso and group NGC estimates remain unchanged it is evident that the performances of all the estimators are affected by the presence of one large group, skewing the uniform nature of the network. For example the MCC measures of group NGC estimates in a balanced network with $p = 60$ and $|E| = 351$ vary around $97 - 98\%$ which lowers to $89\% - 90\%$ when the groups are unbalanced.

The results for misspecified groups are given in Table 3. Note that for higher sample size n, the MCC of lasso and regular group lasso are comparable. However, the thresholded version of group lasso achieves significantly higher MCC than the rest. This demonstrates the advantage of using the directional consistency of group lasso estimators to perform

		$p = 60$, $\|E\| = 450$ Groups=$1 \times 10, 11 \times 5$			$p = 120$, $\|E\| = 1575$ Groups=$1 \times 10, 23 \times 5$			$p = 200$, $\|E\| = 4150$ Groups=$1 \times 10, 39 \times 5$		
	n	160	110	60	160	110	60	160	110	60
P	Lasso	72(2)	69(3)	62(2)	51(1)	48(1)	41(1)	61(1)	53(1)	42(2)
	Grp	84(4)	79(6)	76(9)	55(5)	47(5)	40(6)	86(3)	77(5)	66(7)
	Thgrp	86(4)	82(7)	78(11)	60(6)	50(7)	40(5)	88(2)	79(6)	69(6)
	Agrp	85(3)	81(5)	77(9)	59(5)	51(5)	42(6)	88(2)	78(5)	67(6)
R	Lasso	45(2)	35(2)	22(2)	43(1)	34(1)	22(1)	23(1)	15(0)	7(0)
	Grp	94(3)	87(5)	61(8)	88(2)	75(5)	48(6)	73(3)	49(6)	22(5)
	Thgrp	95(2)	88(4)	62(8)	89(3)	77(4)	50(5)	73(3)	50(6)	21(5)
	Agrp	94(3)	87(5)	61(8)	88(2)	75(5)	48(6)	73(3)	49(6)	22(5)
MCC	Lasso	56(2)	48(2)	35(2)	46(1)	39(1)	29(1)	36(1)	28(1)	17(1)
	Grp	89(3)	82(4)	67(5)	68(3)	58(3)	42(3)	79(1)	61(3)	37(3)
	Thgrp	90(3)	84(4)	68(6)	72(4)	61(4)	43(2)	80(1)	62(3)	37(3)
	Agrp	89(3)	83(4)	67(6)	71(3)	60(3)	43(3)	79(1)	61(3)	37(3)
ERR	Lasso	10.59	10.74	11.76	18.3	18.72	18.76	11.54	10.93	9.29
LAG	Grp	7.04	9.85	13.04	12.53	14.71	13.06	4.8	6.41	6.85
	Thgrp	6.58	8.98	11.1	9.6	11.9	10.9	4.06	5.65	5.7
	Agrp	6.74	9.19	12.96	10.81	12.78	11.79	4.55	6.2	6.81

Table 2: Performance of different regularization methods in estimating graphical Granger causality with **unbalanced** group sizes and no misspecification; $d = 2$, $T = 5$, $SNR = 1.8$. Precision (P), Recall (R), MCC are given in percentages (numbers in parentheses give standard deviations). ERR LAG gives the error associated with incorrect estimation of VAR order.

		$p = 60$, $\|E\| = 246$ Group Size=6			$p = 120$, $\|E\| = 968$ Group Size=6		
	n	160	110	60	160	110	60
P	Lasso	88(2)	85(3)	77(5)	59(1)	55(1)	49(2)
	Grp	65(2)	66(2)	66(3)	43(3)	44(4)	38(4)
	Thgrp	87(3)	88(3)	85(3)	56(6)	56(6)	51(7)
	Agrp	65(2)	66(2)	66(3)	45(2)	45(4)	39(4)
R	Lasso	80(3)	63(3)	37(2)	66(1)	54(1)	35(1)
	Grp	100(0)	98(2)	82(6)	87(2)	78(3)	59(4)
	Thgrp	100(0)	98(2)	79(6)	86(2)	79(3)	57(4)
	Agrp	100(0)	98(2)	82(6)	86(2)	78(3)	58(3)
MCC	Lasso	84(2)	73(2)	53(3)	62(1)	54(1)	41(1)
	Grp	81(1)	80(2)	74(4)	61(2)	58(3)	47(2)
	Thgrp	93(2)	93(2)	82(4)	69(4)	66(4)	53(3)
	Agrp	81(1)	80(2)	74(4)	62(2)	59(2)	47(2)
ERR	Lasso	12.63	17.05	22.41	45.09	49.68	53.4
LAG	Grp	9.43	8.78	15.12	18.22	18.43	29.26
	Thgrp	6.45	5.34	8.02	11.81	12.84	15.57
	Agrp	9.11	8.78	14.96	16.32	16.9	27.69

Table 3: Performance of different regularization methods in estimating graphical Granger causality with **misspecified** groups (30% misspecification); $d = 2$, $T = 10$, $SNR = 2$. Precision (P), Recall (R), MCC are given in percentages (numbers in parentheses give standard deviations). ERR LAG gives the error associated with incorrect estimation of VAR order.

	Lasso	Grp	Agrp	Thgrp
mean	0.649	0.456	0.457	0.456
stdev	0.340	0.252	0.251	0.252

Table 4: Mean and standard deviation of MSE for different NGC estimates

within group variable selection. We would like to mention here that a careful choice of the thresholding parameters δ_{grp} and $\delta_{misspec}$ via cross-validation improves the performance of thresholded group lasso; however, we do not pursue these methods here as they require grid search over many tuning parameters or an efficient estimator of the degree of freedom of group lasso.

In summary, the results clearly show that all variants of group lasso NGC outperform the lasso-based ones, whenever the grouping structure of the variables is known and correctly specified. Further, their performance depends on the composition of group sizes. On the other hand, if the a priori known group structure is moderately misspecified lasso estimates produce comparable results to regular and adaptive group NGC ones, while thresholded group estimates outperform all other methods, as expected.

6. Application

Example: T-cell activation. Estimation of gene regulatory networks from expression data is a fundamental problem in functional genomics (Friedman, 2004). Time course data coupled with NGC models are informationally rich enough for the task at hand. The data for this application come from Rangel et al. (2004), where expression patterns of genes involved in T-cell activation were studied with the goal of discovering regulatory mechanisms that govern them in response to external stimuli. Activated T-cells are involved in regulation of effector cells (e.g., B-cells) and play a central role in mediating immune response. The available data comprising of $n = 44$ samples of $p = 58$ genes, measure the cells response at 10 time points, $t = 0, 2, 4, 6, 8, 18, 24, 32, 48, 72$ hours after their stimulation with a T-cell receptor independent activation mechanism. We concentrate on data from the first 5 time points, that correspond to early response mechanisms in the cells.

Genes are often grouped based on their function and activity patterns into biological pathways. Thus, the knowledge of gene functions and their membership in biological pathways can be used as inherent grouping structures in the proposed group lasso estimates of NGC. Towards this, we used available biological knowledge to define groups of genes based on their biological function. Reliable information for biological functions were found from the literature for 38 genes, which were retained for further analysis. These 38 genes were grouped into 13 groups with the number of genes in different groups ranging from 1 to 5.

Figure 5 shows the estimated networks based on lasso and thresholded group lasso estimates, where for ease of representation the nodes of the network correspond to groups of genes. In this case, estimates from variants of group NGC estimator were all similar, and included a number of known regulatory mechanisms in T-cell activation, not present in the regular lasso estimate. For instance, Waterman et al. (1990) suggest that TCF plays a significant role in activation of T-cells, which may describe the dominant role of this group of genes in the activation mechanism. On the other hand, Kim et al. (2005) suggest that

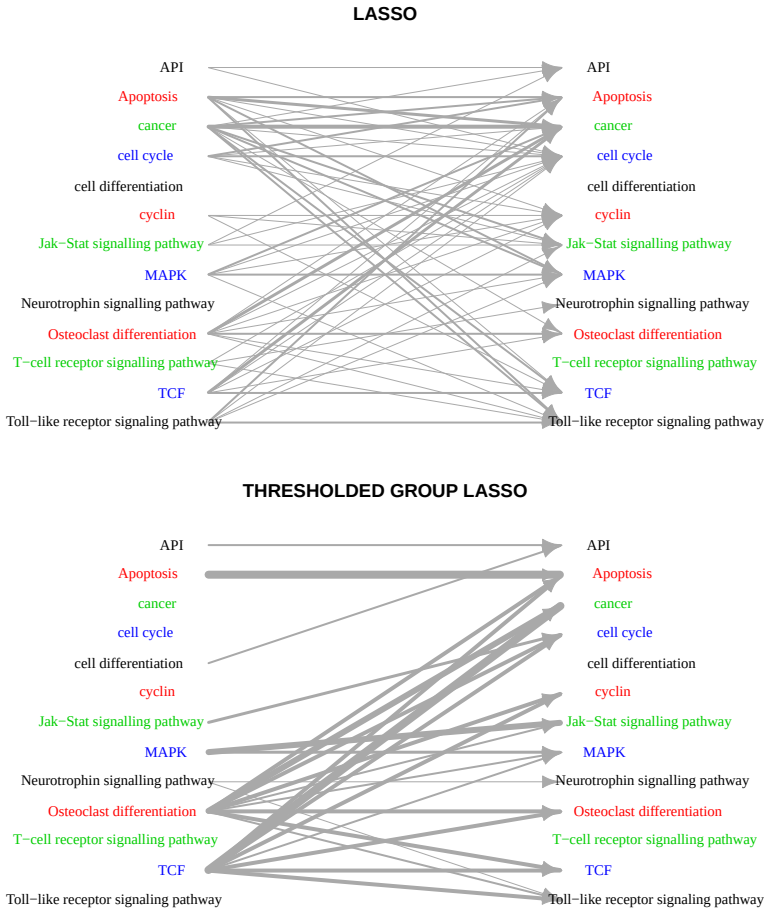


Figure 5: Estimated Gene Regulatory Networks of T-cell activation. Width of edges represent the number of effects between two groups, and the network represents the aggregated regulatory network over 3 time points.

activated T-cells exhibit high levels of osteoclast-associated receptor activity which may attribute the large number of associations between member of osteoclast differentiation and other groups. Finally, the estimated networks based on variants of group lasso estimator also offer improved estimation accuracy in terms of mean squared error (MSE) despite having having comparable complexities to their regular lasso counterpart (Table 4), which further confirms the findings of other numerical studies in that paper.

Example: Banking balance sheets application. In this application, we examine the structure of the balance sheets in terms of assets and liabilities of the $n = 50$ largest (in terms of total balance sheet size) US banking corporations. The data cover 9 quarters (September 2009-September 2011) and were directly obtained from the Federal Deposit Insurance Corporation (FDIC) database (available at `www.fdic.gov`). The $p = 21$ variables

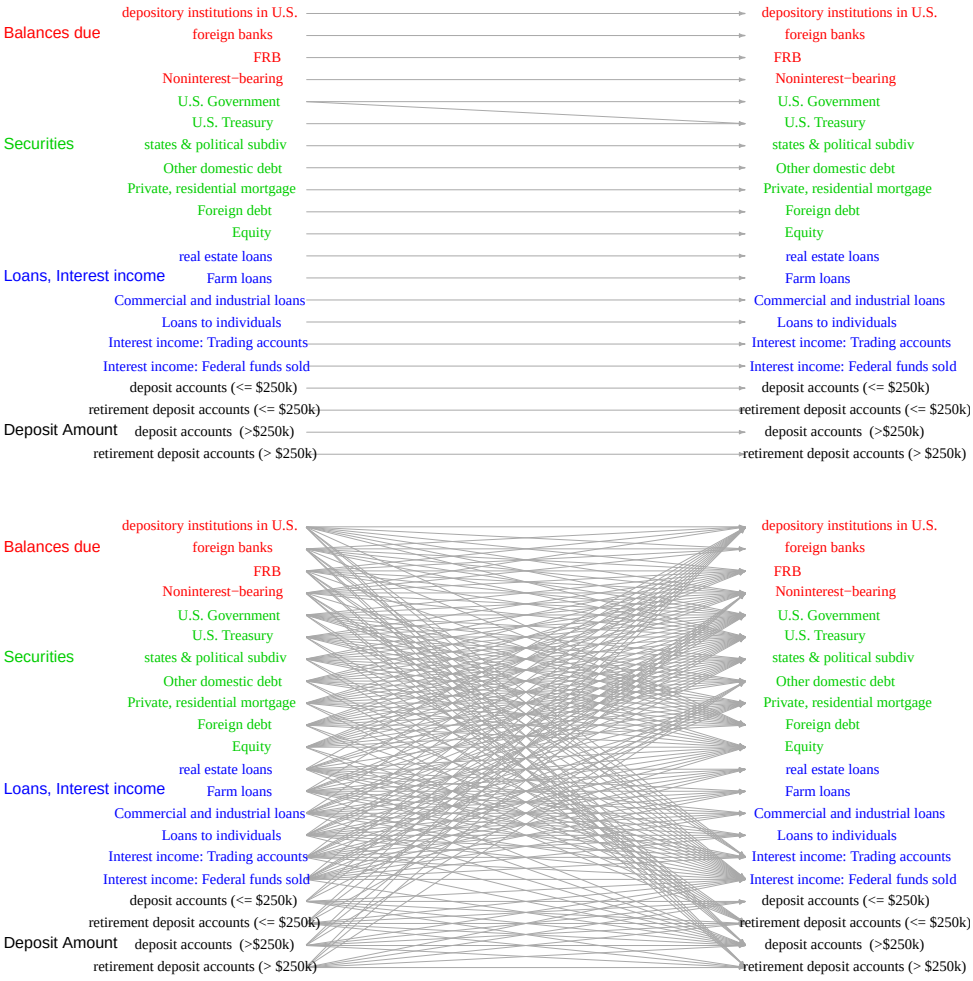


Figure 6: Estimated Networks of banking balance sheet variables using (a) lasso and (b) group lasso. The networks represent the aggregated network over 5 time points.

correspond to different assets (US and foreign government debt securities, equities, loans (commercial, mortgages), leases, etc.) and liabilities (domestic and foreign deposits from households and businesses, deposits from the Federal Reserve Board, deposits of other financial institutions, non-interest bearing liabilities, etc.) We have organized them into four categories: two for the assets (loans and securities) and two for the liabilities (Balances Due and Deposits, based on a $250K reporting FDIC threshold). Amongst the 50 banks examined, one discerns large integrated ones with significant retail, commercial and investment activities (e.g., Citibank, JP Morgan, Bank of America, Wells Fargo), banks primarily focused on investment business (e.g., Goldman Sachs, Morgan Stanley, American Express, E-Trade, Charles Schwab), regional banks (e.g., Banco Popular de Puerto Rico, Comerica Bank, Bank of the West).

Quarter	Lasso	Grp	Agrp	Thgrp
Dec 2010	1.59 (0.29)	0.36 (0.05)	0.36 (0.05)	0.37 (0.05)
Mar 2011	1.46 (0.30)	0.47 (0.23)	0.47 (0.23)	0.46 (0.22)
Jun 2011	1.33 (0.26)	0.36 (0.11)	0.36 (0.11)	0.35 (0.11)
Sep 2011	1.72 (0.32)	0.50 (0.18)	0.50 (0.18)	0.47 (0.16)

Table 5: Mean and standard deviation (in parentheses) of PMSE (MSE in case of Dec 2010) for prediction of banking balance sheet variables.

The raw data are reported in thousands of dollars. The few missing values were imputed using a nearest neighbor imputation method with $k = 5$, by clustering them according to their total assets in the most recent quarter in the data collection period (September 2011) and subsequently every missing observation for a particular bank was imputed by the median observation on its five nearest neighbors. The data were log-transformed to reduce non-stationarity issues. The data set was restructured as a panel with $p = 21$ variables and $n = 50$ replicates observed over $T = 9$ time points. Every column of replicates was scaled to have unit variance.

We applied the proposed variants of NGC estimates on the first $T = 6$ time points (Sep 2009 - Dec 2010) of the above panel data set. The parameters λ and δ_{grp} were chosen using a $19 : 1$ sample-splitting method and the misspecification threshold $\delta_{misspec}$ was set to zero as the grouping structure was reliable. We calculated the MSE of the fitted model in predicting the outcomes in the four quarters (December 2010 - September 2011). The Predicted MSE (MSE for Dec 2010) are listed in Table 5. The estimated network structures are shown in Figure 6.

It can be seen that the lasso estimates recover a very simple temporal structure amongst the variables; namely, that past values (in this case lag-1) influence present ones. Given the structure of the balance sheet of large banks, this is an anticipated result, since it can not be radically altered over a short time period due to business relationships and past commitments to customers of the bank. However, the (adaptive) group lasso estimates reveal a richer and more nuanced structure. Examining the fitted values of the adjacency matrices A^t, we notice that the dominant effects remain those discovered by the lasso estimates. However, fairly strong effects are also estimated within each group, but also between the groups of the assets (loans and securities) on the balance sheet. This suggests rebalancing of the balance sheet for risk management purposes between relatively low risk securities and potentially more risky loans. Given the period covered by the data (post financial crisis starting in September 2009) when credit risk management became of paramount importance, the analysis picks up interesting patterns. On the other hand, significant fewer associations are discovered between the liabilities side of the balance sheet. Finally, there exist relationships between deposits and securities such as US Treasuries and other domestic ones (primarily municipal bonds); the latter indicates that an effort on behalf of the banks to manage the credit risk of their balance sheets, namely allocating to low risk assets as opposed to more risky loans.

It is also worth noting that the group lasso model exhibits superior predictive performance over the lasso estimates, even 4 quarters into the future. Finally, in this case the

thresholded estimates did not provide any additional benefits over the regular and adaptive variants, given that the specification of the groups was based on accounting principles and hence correctly structured.

7. Discussion

In this paper, the problem of estimating Network Granger Causal (NGC) models with inherent grouping structure is studied when replicates are available. Norm, and both group level and within group variable selection consistency are established under fairly mild assumptions on the structure of the underlying time series. To achieve the second objective the novel concept of direction consistency is introduced.

The type of NGC models discussed in this study have wide applicability in different areas, including genomics and economics. However, in many contexts the availability of replicates at each time point is not feasible (e.g., in rate of returns for stocks or other macroeconomic variables), while grouping structure is still present (e.g., grouping of stocks according to industry sector). Hence, it is of interest to study the behavior of group lasso estimates in such a setting and address the technical challenges emanating from such a pure time series (dependent) data structure.

Acknowledgments

We thank the action editor and three anonymous reviewers for their helpful comments. The work of SB and GM was supported in part by DoD grant W81XWH-12-1-0130, and that of GM by NSF DMS-1106695 and NSA H98230-10-1-0203. The work of AS was partially supported by NSF grant DMS-1161565 and NIH grant 1R21GM101719-01A1.

Appendix A. Auxiliary Lemmas

Lemma A.1 (Characterization of the Group lasso estimate) *A vector $\hat{\beta} \in \mathbb{R}^p$ is a solution to the convex optimization problem*

$$\underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \frac{1}{2n} \|Y - X\beta\|^2 + \sum_{g=1}^{G} \lambda_g \|\beta_{[g]}\| \tag{14}$$

if and only if $\hat{\beta}$ satisfies, for some $\tau \in \mathbb{R}^p$ with $\max_{1 \le g \le G} \left\|\tau_{[g]}\right\| \le 1$, $\frac{1}{n}\left[X'(Y - X\hat{\beta})\right]_{[g]} = \lambda_g \tau_{[g]} \, \forall g$. Further, $\tau_{[g]} = D\left(\hat{\beta}_{[g]}\right)$ whenever $\hat{\beta}_{[g]} \neq \mathbf{0}$.

Proof Follows directly from the KKT conditions for the optimization problem (14). ■

Lemma A.2 (Concentration bound for multivariate Gaussian) *Let $Z_{k\times 1} \sim N(0, \Sigma)$. Then, for any $t > 0$, the following inequalities hold:*

$$\mathbb{P}\left(\|\|Z\| - \mathbb{E}\|Z\|\| > t\right) \le 2 \exp\left(-\frac{2t^2}{\pi^2 \|\Sigma\|}\right), \qquad \mathbb{E}\|Z\| \le \sqrt{k}\sqrt{\|\Sigma\|}.$$

Proof The first inequality can be found in Ledoux and Talagrand (1991) (equation (3.2). To establish the second inequality note that,

$$\mathbb{E}\|Z\| \leq \sqrt{\mathbb{E}\|Z\|^2} = \sqrt{\mathbb{E}\left[\mathrm{tr}\left(ZZ'\right)\right]} = \sqrt{\mathrm{tr}\left(\Sigma\right)} \leq \sqrt{k}\sqrt{\|\Sigma\|}.$$

■

Lemma A.3 *Let $\beta,\ \hat{\beta} \in \mathbb{R}^m \backslash \{\mathbf{0}\}$. Let $\hat{u} = \hat{\beta} - \beta$ and $r = D(\hat{\beta}) - D(\beta)$. Then $\|r\| < 2\delta$ whenever $\|\hat{u}\| < \delta \|\beta\|$.*

Proof It follows from $\|\hat{u}\| < \delta \|\beta\|$ that

$$(1-\delta)\|\beta\| < \|\beta\| - \|\hat{u}\| \leq \|\hat{\beta}\| \leq \|\hat{u}\| + \|\beta\| < (1+\delta)\|\beta\|\,,$$

which implies that $\left|\|\beta\| - \|\hat{\beta}\|\right| < \delta\|\beta\|$. Now,

$$\|\hat{\beta}\|\,\|\beta\|\|r\| = \left\|\ \hat{\beta}\|\beta\| + (\hat{u}-\hat{\beta})\|\hat{\beta}\|\ \right\| \leq \left\|\hat{\beta}\left(\|\beta\| - \|\hat{\beta}\|\right) + \|\hat{\beta}\|\ \hat{u}\ \right\| < \|\hat{\beta}\|\,\|\beta\|(\delta+\delta),$$

since $\left|\|\beta\| - \|\hat{\beta}\|\right| < \delta\|\beta\|$ and $\|\hat{u}\| < \delta\,\|\beta\|$. ■

Lemma A.4 *Let $\mathcal{G}_1, \ldots, \mathcal{G}_G$ be any partition of $\{1, \ldots, p\}$ into G non-overlapping groups and $\lambda_1, \ldots, \lambda_G$ be positive real numbers. Define the cone sets $\mathcal{C}(J, L) = \{v \in \mathbb{R}^p : \sum_{g \notin J} \lambda_g \|v_{[g]}\| \leq L \sum_{g \in J} \lambda_g \|v_{[g]}\|\}$ for any subset of groups $J \subseteq \mathbb{N}_G$. Also define the set of group s-sparse vectors $\mathbb{D}(s) := \{v \in \mathbb{R}^p : \|v\| \leq 1,\ supp(v) \subseteq \mathcal{G}_J$ for some $J \subseteq \mathbb{N}_G,\ |J| \leq s\}$. Then*

$$\bigcup_{J \subseteq \mathbb{N}_G, |J| \leq s} \mathcal{C}(J, L) \cap \mathbb{S}^{p-1} \subseteq (2 + L')cl\{conv\{\mathbb{D}(s)\}\}, \tag{15}$$

where $L' = L\lambda_{\max}/\lambda_{\min}$, $\mathbb{S}^{p-1} = \{v \in \mathbb{R}^p : \|v\| = 1\}$ is the ball of unit norm vectors in $\mathbb{R}^p$ and $cl\{.\}$, $conv\{.\}$ respectively denote the closure and convex hull of a set.

Proof Note that for any $J \subseteq \mathbb{N}_G$, $|J| \leq s$, and $v \in \mathcal{C}(J, L) \cap \mathbb{S}^{p-1}$, we have

$$\sum_{g \notin J} \|v_{[g]}\| \leq L \frac{\lambda_{\max}}{\lambda_{\min}} \sum_{g \in J} \|v_{[g]}\|,$$

which implies

$$\|v\|_{2,1} \leq (L'+1) \sum_{g \in J} \|v_{[g]}\| \leq (L'+1)\sqrt{s}\|v_{[J]}\| \leq (L'+1)\sqrt{s}.$$

Hence the union of the cone sets on the left hand side of (15) is a subset of $A := \{v \in \mathbb{R}^p : \|v\| \leq 1,\ \|v\|_{2,1} \leq (L'+1)\sqrt{s}\}$.

We will show that the set A is a subset of $B := (2 + L')cl\{conv\{\mathbb{D}(s)\}\}$, the closed convex hull on the right hand side of (15). Since both sets A and B are closed convex, it is enough to show that the support function of A is dominated by the support function of B.

The support function of A is given by $\phi_A(z) = \sup_{\theta \in A}\langle \theta, z\rangle$. For any $z \in \mathbb{R}^p$, let $S \subseteq \{1, \ldots, G\}$ be a subset of top s groups in terms of the ℓ_2 norm of $z_{[g]}$. Thus, $\|z_{[S^c]}\|_{2,\infty} \leq \|z_{[g]}\|$ for all $g \in S$. This implies $\|z_{[S^c]}\|_{2,\infty} \leq (1/s)\|z_{[S]}\|_{2,1} \leq (1/\sqrt{s})\|z_{[S]}\|$. So, we have

$$\begin{aligned} \phi_A(z) = \sup_{\theta \in A}\langle \theta, z\rangle &\leq \sup_{\|\theta_{[S]}\| \leq 1} \langle \theta_{[S]}, z_{[S]}\rangle + \sup_{\|\theta_{[S^c]}\|_{2,1} \leq \sqrt{s}(L'+1)} \langle \theta_{[S^c]}, z_{[S^c]}\rangle \qquad (16) \\ &\leq \|z_{[S]}\| + (L'+1)\sqrt{s}\|z_{[S^c]}\|_{2,\infty} \leq (L'+2)\|z_{[S]}\|. \qquad (17) \end{aligned}$$

On the other hand, support function of $B := (L'+2)cl\{conv\{\mathbb{D}(s)\}\}$ is given by

$$\phi_B(z) = \sup_{\theta \in B}\langle \theta, z\rangle = (L'+2) \max_{|U|=s,\ U \subseteq \mathbb{N}_G} \sup_{\|\theta_{[U]}\| \leq 1} \langle \theta_{[U]}, z_{[U]}\rangle = (L'+2)\|z_{[S]}\|.$$

This concludes the proof. ■

Lemma A.5 *Consider a matrix $X_{n\times p}$ with rows independently distributed as $N(0, \Sigma)$, $\Lambda_{\min}(\Sigma) > 0$. Let $\mathcal{G}_1, \ldots, \mathcal{G}_G$ be any partition of $\{1, \ldots, p\}$ into G non-overlapping groups of size $k_1, \ldots, k_g$, respectively. Let $C = X'X/n$ denote the sample Gram matrix and $\mathbb{D}(s)$ denote the set of group s-sparse vectors defined in Lemma A.4. Then, for any integer $s \geq 1$ and any $\eta > 0$, we have*

$$\begin{aligned} &\mathbb{P}\left[\sup_{v \in cl\{conv\{\mathbb{D}(s)\}\}} |v'(C - \Sigma)v| > 6\eta\|\Sigma\|\right] \\ \leq\ & c_0 \exp\left[-n\min\{\eta, \eta^2\} + c_1 s(k_{\max} + c_2 \log(eG/2s))\right] \qquad (18) \end{aligned}$$

for some universal positive constants c_i.

Proof We consider a fixed vector $v \in \mathbb{R}^p$ with $\|v\| \leq 1$, the support of which can be covered by a set J of at most s groups, i.e., $supp(v) \subseteq \mathcal{G}_J$, $J \subseteq \mathbb{N}_G$, $|J| \leq s$. Define $Y = Xv$. Then each coordinate of Y is independently distributed as $N(0, \sigma_y^2)$, where $\sigma_y^2 = v'\Sigma v \leq \|\Sigma\|$.

Then, for any $\eta > 0$, Hanson-Wright inequality of Rudelson and Vershynin (2013) ensures

$$\mathbb{P}\left[\left|v'(C - \Sigma)v\right| > \eta\|\Sigma\|\right] \leq \mathbb{P}\left[\frac{1}{n}\left|Y'Y - \mathbb{E}Y'Y\right| > \eta\sigma_y^2\right] \leq 2\exp\left[-cn\min\{\eta, \eta^2\}\right].$$

Next, we extend this deviation bound on all vectors v in the sparse set

$$\mathbb{D}(2s) = \{v \in \mathbb{R}^p : \|v\| \leq 1,\ supp(v) \subseteq \mathcal{G}_J \text{ for some } J \subseteq \mathbb{N}_G,\ |J| \leq 2s\}. \qquad (19)$$

For a given $J \subseteq \mathbb{N}_G$, $|J| = 2s$, we define $\mathbb{D}_J = \{v \in \mathbb{R}^p : \|v\| \leq 1, supp(v) \subseteq \mathcal{G}_J\}$ and note that $\mathbb{D}(2s) = \cup_{|J|=2s}\mathbb{D}_J$. For an $\epsilon > 0$ to be specified later, we construct an ϵ-net $\mathcal{A}$ of $\mathbb{D}_J$. Since $\sum_{g \in J} k_g \leq 2s\, k_{\max}$, it is possible to construct such a net $\mathcal{A}$ with cardinality at most $(1 + 2/\epsilon)^{2s\, k_{\max}}$ (Vershynin, 2009).

We want a tail inequality for $M := \sup_{v \in \mathbb{D}_J} |v'\Delta v|$, where $\Delta = C - \Sigma$. Since $\mathcal{A}$ is an ϵ-cover of $\mathbb{D}_J$, for any $v \in \mathbb{D}_J$, there exists $v_0 \in \mathcal{A}$ such that $w = v - v_0$ satisfies $\|w\| \leq \epsilon$. Then

$$|v'\Delta v| = |(w + v_0)'\Delta(w + v_0)| \leq |w'\Delta w| + |v_0'\Delta v_0| + 2|v_0'\Delta w|.$$

Taking supremum over all $v \in \mathbb{D}_J$, and noting that $w/\epsilon \in \mathbb{D}_J$, we obtain

$$M \leq \epsilon^2 M + \max_{v_0 \in \mathcal{A}} |v_0'\Delta v_0| + \sup_{u,v \in \mathbb{D}_J} 2\epsilon |u'\Delta v|. \tag{20}$$

To upper bound the third term, note that $(u + v)/2 \in \mathbb{D}_J$, and

$$2|u'\Delta v| \leq |(u + v)'\Delta(u + v)| + |u'\Delta u| + |v'\Delta v|.$$

Hence

$$\sup_{u,v \in \mathbb{D}_J} 2\epsilon |u'\Delta v| \leq 4\epsilon M + \epsilon M + \epsilon M = 6\epsilon M.$$

From equation (20), we now have

$$M \leq (1 - 6\epsilon - \epsilon^2)^{-1} \max_{v_0 \in \mathcal{A}} |v_0'\Delta v_0|.$$

Choosing $\epsilon > 0$ small enough so that $(1 - 6\epsilon - \epsilon^2) > 1/2$, we obtain

$$\begin{aligned} \mathbb{P}\left[\sup_{v \in \mathbb{D}_J} |v'\Delta v| > 2\eta\|\Sigma\|\right] &\leq \mathbb{P}\left[\max_{v_0 \in \mathcal{A}} |v_0'\Delta v_0| > \eta\|\Sigma\|\right] \\ &\leq 2\,(1 + 2/\epsilon)^{2s\,k_{\max}} \exp[-cn\min\{\eta, \eta^2\}]. \end{aligned}$$

Taking supremum over $\begin{pmatrix} G \\ 2s \end{pmatrix} \leq (eG/2s)^{2s}$ choices of J, we get

$$\mathbb{P}\left[\sup_{v \in \mathbb{D}(2s)} |v'\Delta v| > 2\eta\|\Sigma\|\right] \leq 2\exp\left[-cn\min\{\eta, \eta^2\} + 2s\log\left(\frac{eG}{2s}\right) + 2s\,k_{\max}\log\left(1 + \frac{2}{\epsilon}\right)\right].$$

In order to extend this deviation inequality to $cl\{conv\{\mathbb{D}(s)\}\}$, we note that any v in the convex hull of $\mathbb{D}(s)$ can be expressed as $v = \sum_{i=1}^m \alpha_i v_i$, where $v_1, \ldots, v_m$ are in $\mathbb{D}(s)$ and $0 \leq \alpha_i \leq 1$, $\sum \alpha_i = 1$. Then

$$|v'\Delta v| \leq \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j |v_i'\Delta v_j|.$$

Also, for every $i,\ j$, $(v_i + v_j)/2 \in \mathbb{D}(2s)$, and

$$|v_i'\Delta v_j| \leq \frac{1}{2}\left[|(v_i + v_j)'\Delta(v_i + v_j)| + |v_i'\Delta v_i| + |v_j'\Delta v_j|\right].$$

Hence

$$\sup_{v \in conv\{\mathbb{D}(s)\}} |v'\Delta v| \leq \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j\ \frac{1}{2}[4 + 1 + 1] \sup_{v \in \mathbb{D}(2s)} |v'\Delta v|.$$

Together with the continuity of quadratic forms, this implies

$$\sup_{v \in cl\{conv\{\mathbb{D}(s)\}\}} |v'\Delta v| \leq 3 \sup_{v \in \mathbb{D}(2s)} |v'\Delta v|.$$

The result then readily follows from the above deviation inequality. ■

Appendix B. Proof of Main Results

Proof [Proof of Proposition 3.2] (a) Note that Σ is a $p(T-1) \times p(T-1)$ block Toeplitz matrix with $(i,j)^{th}$ block $(\Sigma_{ij})_{1 \leq i,j \leq (T-1)} := \Gamma(i-j)$, where $\Gamma(\ell)_{p \times p}$ is the autocovariance function of lag ℓ for the zero-mean VAR(d) process (2), defined as $\Gamma(\ell) = \mathbb{E}[\mathbf{X}^t(\mathbf{X}^{\mathbf{t}-\ell})']$.

We consider the cross spectral density of the VAR(d) process (2)

$$f(\theta) = \frac{1}{2\pi} \sum_{\ell=-\infty}^{\infty} \Gamma(\ell) e^{-i\ell\theta}, \quad \theta \in [-\pi, \pi]. \tag{21}$$

From standard results of spectral theory we know that $\Gamma(\ell) = \int_{-\pi}^{\pi} e^{i\ell\theta} f(\theta)\, d\theta$, for every ℓ.

We want to find a lower bound on the minimum eigenvalue of Σ, i.e., $\inf_{\|x\|=1} x'\Sigma x$. Consider an arbitrary $p(T-1)$-variate unit norm vector x, formed by stacking the p-tuples $x^1, \ldots, x^{T-1}$.

For every $\theta \in [-\pi, \pi]$, define $G(\theta) = \sum_{t=1}^{T-1} x^t\, e^{-it\theta}$ and note that

$$\begin{aligned}
\int_{-\pi}^{\pi} G^*(\theta) G(\theta)\, d\theta &= \sum_{t=1}^{T-1} \sum_{\tau=1}^{T-1} (x^t)'(x^\tau) \int_{-\pi}^{\pi} e^{i(t-\tau)\theta}\, d\theta \\
&= \sum_{t=1}^{T-1} \sum_{\tau=1}^{T-1} (x^t)'(x^\tau)\ (2\pi\, \mathbf{1}_{\{t=\tau\}}) = 2\pi \sum_{t=1}^{T-1} (x^t)'(x^t) = 2\pi\ \|x\|^2 = 2\pi.
\end{aligned}$$

Also let $\mu(\theta)$ be the minimum eigenvalue of the Hermitian matrix $f(\theta)$. Following Parter (1961) we have the result

$$\begin{aligned}
x'\Sigma x &= \sum_{t=1}^{T-1} \sum_{\tau=1}^{T-1} (x^t)'\Gamma(t-\tau)x^\tau = \sum_{t=1}^{T-1} \sum_{\tau=1}^{T-1} (x^t)' \left(\int_{-\pi}^{\pi} e^{i(t-\tau)\theta} f(\theta) d\theta \right) x^\tau \\
&= \int_{-\pi}^{\pi} \left(\sum_{t=1}^{T-1} (x^t)' e^{it\theta} \right) f(\theta) \left(\sum_{\tau=1}^{T-1} x^\tau e^{-i\tau\theta} \right) d\theta = \int_{-\pi}^{\pi} G^*(\theta)\, f(\theta)\, G(\theta)\, d\theta \\
&\geq \int_{-\pi}^{\pi} \mu(\theta)\, (G^*(\theta)G(\theta))\ d\theta \geq \left(\min_{\theta \in (-\pi,\pi)} \mu(\theta) \right) \int_{-\pi}^{\pi} G^*(\theta) G(\theta)\, d\theta = 2\pi \min_{\theta \in (-\pi,\pi)} \mu(\theta).
\end{aligned}$$

So $\Lambda_{min}(\Sigma) \geq 2\pi \min_{\theta \in (-\pi,\pi)} \mu(\theta)$. Since $\mathcal{A}(z) = I - A^1 z - A^2 z^2 - \ldots - A^d z^d$ is the (matrix-valued) characteristic polynomial of the VAR(d) model (2), we have the following representation of the spectral density (see Priestley, 1981, eqn 9.4.23):

$$f(\theta) = \frac{1}{2\pi} \sigma^2 (\mathcal{A}(e^{-i\theta}))^{-1} (\mathcal{A}^*(e^{-i\theta}))^{-1}.$$

Thus, $2\pi\mu(\theta) = 2\pi\Lambda_{min}(f(\theta)) = 2\pi/\Lambda_{max}(f(\theta)^{-1}) \geq \sigma^2/\left\|\mathcal{A}(e^{-i\theta})\right\|^2$. But $\left\|\mathcal{A}(e^{-i\theta})\right\| \leq 1 + \sum_{t=1}^{d}\left\|A^t\right\|$ for every $\theta \in [-\pi, \pi]$. The result then follows at once from the standard matrix norm inequality (see e.g., Golub and Van Loan, 1996, Cor 2.3.2)

$$\|A^t\|_2 \leq \sqrt{\|A^t\|_1\|A^t\|_\infty} \leq \frac{\|A^t\|_1 + \|A^t\|_\infty}{2} \quad t = 1, \ldots, d,$$

where

$$\|A^t\|_1 = \max_{1\leq i\leq p}\sum_{j=1}^{p}|A^t_{ij}|, \quad \|A^t\|_\infty = \max_{1\leq j\leq p}\sum_{i=1}^{p}|A^t_{ij}|.$$

(b) The first part of the proposition ensures that $\Lambda_{min}(\Sigma) \geq \sigma^2\left[1 + \frac{1}{2}(\mathbf{v}_{in} + \mathbf{v}_{out})\right]^{-2}$. If the replicates available from different panels are i.i.d, each row of the design matrix is independently and identically distributed according to a $N(\mathbf{0}, \Sigma)$ distribution.

To show that RE(s, L) of (5) holds with high probability for sufficiently large n, it is enough to show that

$$\min_{\substack{v\in\mathcal{C}(J,L)\setminus\{0\}\\ J\subset\mathbb{N}_{\bar{G}},\ |J|\leq s}} \frac{1}{n}\frac{\|\mathbf{X}v\|^2}{\|v\|^2} \geq \phi^2_{RE} \tag{22}$$

holds with high probability, where the cone sets $\mathcal{C}(J, L)$ are defined as

$$\mathcal{C}(J, L) := \{v \in \mathbb{R}^{\bar{p}} : \sum_{g\notin J}\lambda_g\|v_{[g]}\| \leq L\sum_{g\in J}\lambda_g\|v_{[g]}\|\} \tag{23}$$

for all $J \subset \mathbb{N}_{\bar{G}}$ with $|J| \leq s$. Denote the ball of unit norm vectors in $\mathbb{R}^{\bar{p}}$ by $\mathbb{S}^{\bar{p}-1}$. By scale invariance of $\|\mathbf{X}v\|^2/n\|v\|^2$, it is enough to show that with high probability

$$\min_{\substack{v\in\mathbb{S}^{\bar{p}-1}\cap\mathcal{C}(J,L)\\ J\subset\mathbb{N}_{\bar{G}},\ |J|\leq s}} v'Cv \geq \phi^2_{RE}, \tag{24}$$

where $C = \mathbf{X}'\mathbf{X}/n$ is the sample Gram matrix.

By part (a), we already know that $v'\Sigma v \geq \Lambda_{\min}(\Sigma) > 0$ for all $v \in \mathbb{S}^{\bar{p}-1}$. So we only need to show that $|v'(C - \Sigma)v| \leq \Lambda_{\min}(\Sigma)/2$ with high probability, uniformly on the set

$$\bigcup_{J\subseteq\mathbb{N}_{\bar{G}},|J|\leq s}\mathcal{C}(J, L)\cap\mathbb{S}^{\bar{p}-1}. \tag{25}$$

The proof relies on two key parts. In the first part, we use an extremal representation to show that the above union of the cone sets sits within the closed convex hull of a suitably defined set of group s-sparse vectors. In particular, it follows from Lemma A.4 that

$$\bigcup_{J\subseteq\mathbb{N}_{\bar{G}},\ |J|\leq s}\mathcal{C}(J, L)\cap\mathbb{S}^{\bar{p}-1} \subseteq (L' + 2)cl\{conv\{\mathbb{D}(s)\}\}, \tag{26}$$

where $\mathbb{D}(s) = \{v \in \mathbb{R}^{\bar{p}} : \|v\| \leq 1,\ supp(v) \subseteq \mathcal{G}_J \text{ for some } J \subseteq \mathbb{N}_{\bar{G}},\ |J| \leq s\}$, $L' = L\lambda_{\max}/\lambda_{\min}$ and $cl\{.\}$, $conv\{.\}$ respectively denote the closure and convex hull of a set.

The next part of the proof is an upper bound on the tail probability of $v'(C-\Sigma)v$, uniformly over all $v \in cl\{conv\{\mathbb{D}(s)\}\}$, presented in Lemma A.5. In particular, setting $\eta = \Lambda_{\min}(\Sigma)/12\|\Sigma\|(2+L')^2$ in the above lemma yields

$$\mathbb{P}\left[\sup_{v\in(2+L')cl\{conv\{\mathbb{D}(s)\}\}} |v'(C-\Sigma)v| > \Lambda_{\min}(\Sigma)/2\right] \leq c_0 \exp[-c_1\, n] \tag{27}$$

for the proposed choice of n. Together with the lower bound on $\Lambda_{\min}(\Sigma)$ established in part (a), this concludes the proof. ■

Proof [Proof of Theorem 4.1] Consider any solution $\hat{\beta}_R \in \mathbb{R}^q$ of the restricted regression

$$\underset{\beta\in\mathbb{R}^q}{\operatorname{argmin}} \frac{1}{2n}\left\|\mathbf{Y} - X_{(1)}\beta\right\|_2^2 + \lambda\sum_{g=1}^{s}\left\|\beta_{[g]}\right\|_2 \tag{28}$$

and set $\hat{\beta} = \left[\hat{\beta}_R' : \mathbf{0}_{1\times(p-q)}\right]'$. We show that such an augmented vector $\hat{\beta}$ satisfies the statements of Theorem 4.1 with high probability.

Let $\hat{u} = \hat{\beta}_{(1)} - \beta^0_{(1)} = \hat{\beta}_R - \beta^0_{(1)}$. In view of lemmas A.1 and A.3, it suffices to show that the following events happen with probability at least $1 - 4\,G^{1-\alpha}$:

$$\left\|\hat{u}_{[g]}\right\| < \delta_n \left\|\beta^0_{[g]}\right\|, \text{ for all } g \in S, \tag{29}$$

$$\frac{1}{n}\left\|\left[X'\left(\epsilon - X_{(1)}\hat{u}\right)\right]_{[g]}\right\| \leq \lambda, \text{ for all } g \notin S. \tag{30}$$

Note that, in view of Lemma A.1, $\hat{u} = (C_{11})^{-1}\left(\frac{1}{\sqrt{n}}Z_{(1)} - \lambda\tau\right)$ for some $\tau \in \mathbb{R}^q$ with $\left\|\tau_{[g]}\right\| \leq 1$ for all $g \in S$, and $Z = \frac{1}{\sqrt{n}}X'\epsilon = \left[Z'_{(1)} : Z'_{(2)}\right]'$. Thus, for any $g \in S$,

$$\begin{aligned}\mathbb{P}\left(\left\|\hat{u}_{[g]}\right\| > \delta_n\left\|\beta^0_{[g]}\right\|\right) &\leq \mathbb{P}\left(\left\|\left[(C_{11})^{-1}\left(\frac{1}{\sqrt{n}}Z_{(1)} - \lambda\tau\right)\right]_{[g]}\right\| > \delta_n\left\|\beta^0_{[g]}\right\|\right)\\ &\leq \mathbb{P}\left(\left\|\left[(C_{11})^{-1}Z_{(1)}\right]_{[g]}\right\| > \sqrt{n}\left[\delta_n\left\|\beta^0{}_{[g]}\right\| - \lambda\left\|\left[(C_{11})^{-1}\tau\right]_{[g]}\right\|\right]\right).\end{aligned}$$

Note that $V = (C_{11})^{-1}Z_{(1)} \sim N(\mathbf{0}, \sigma^2(C_{11})^{-1})$. So $V_{[g]} \sim N(\mathbf{0}, \sigma^2 C_{11}^{[g][g]})$, where $\Sigma^{[g][g]} := (\Sigma^{-1})_{[g][g]}$. Also, by the second statement of lemma A.2 we have $\mathbb{E}\left\|V_{[g]}\right\| \leq \sigma\sqrt{k_g}\sqrt{\left\|C_{11}^{[g][g]}\right\|}$. Therefore $\mathbb{P}\left(\left\|\hat{u}_{[g]}\right\| > \delta_n\left\|\beta^0_{[g]}\right\|\right)$ is bounded above by

$$\begin{aligned}&\mathbb{P}\left(\left|\left\|V_{[g]}\right\| - \mathbb{E}\left\|V_{[g]}\right\|\right| > \sqrt{n}\left[\delta_n\left\|\beta^0{}_{[g]}\right\| - \lambda\left\|(C_{11})^{-1}\right\|\sqrt{s}\right] - \sigma\sqrt{k_g\left\|C_{11}^{[g][g]}\right\|}\right)\\ \leq\, &2\exp\left[-\frac{2}{\pi^2\sigma^2\|C_{11}^{[g][g]}\|}\left(\sqrt{n}\delta_n\|\beta^0_{[g]}\| - \sqrt{n}\lambda\|C_{11}^{-1}\|\sqrt{s} - \sigma\sqrt{k_g\|C_{11}^{[g][g]}\|}\right)^2\right].\end{aligned}$$

For the proposed choice of δ_n, this expression is bounded above by $2\,G^{-\alpha}$.
Next, for any $g \notin S$, we get

$$\mathbb{P}\left(\frac{1}{n}\left\|\left[X'\left(\epsilon - X_{(1)}\hat{u}\right)\right]_{[g]}\right\| > \lambda\right) \leq \mathbb{P}\left(\left\|\left[Z_{(2)} - C_{21}C_{11}^{-1}Z_{(1)}\right]_{[g]}\right\| > \sqrt{n}\lambda\left(1 - \left\|\left[C_{21}C_{11}^{-1}\tau\right]_{[g]}\right\|\right)\right).$$

Defining $W = Z_{(2)} - C_{21}C_{11}^{-1}Z_{(1)} \sim N(\mathbf{0}, \sigma^2(C_{22} - C_{21}C_{11}^{-1}C_{12}))$, the uniform irrepresentable condition implies that the above probability is bounded above by $\mathbb{P}\left(\left\|W_{[g]}\right\| > \sqrt{n}\lambda\eta\right)$.

It can then be seen that $W_{[g]} \sim N(\mathbf{0}, \sigma^2\bar{C}_{[g][g]})$, where $\bar{C} = C_{22} - C_{21}C_{11}^{-1}C_{12}$ denotes the Schur complement of C_{22}. As before, lemma A.2 establishes that

$$\begin{aligned}\mathbb{P}\left(\left\|W_{[g]}\right\| > \sqrt{n}\lambda\eta\right) &\leq \mathbb{P}\left(\left|\left\|W_{[g]}\right\| - \mathbb{E}\left\|W_{[g]}\right\|\right| > \sqrt{n}\lambda\eta - \sigma\sqrt{k_g\|\bar{C}_{[g][g]}\|}\right) \\ &\leq 2\exp\left[-\frac{2}{\pi^2\|\sigma^2\bar{C}_{[g][g]}\|}\left(\sqrt{n}\lambda\eta - \sigma\sqrt{k_g\|\bar{C}_{[g][g]}\|}\right)^2\right],\end{aligned}$$

and the last probability is bounded above by $2G^{-\alpha}$ for the proposed choice of λ.
The results in the proposition follow by considering the union bound on the two sets of the probability statements made across all $g \in \mathbb{N}_G$. ■

Appendix C. Proof of results on ℓ_2-consistency

We first note that each of the p optimization problems in (4) is essentially a generic group lasso regression on n independent samples from a linear model $Y = X\beta^0 + \epsilon$, $\epsilon \sim N(0, \sigma^2)$:

$$\hat{\beta} = \underset{\beta\in\mathbb{R}^p}{\operatorname{argmin}} \frac{1}{2n}\|\mathbf{Y} - \mathbf{X}\beta\|^2 + \sum_{g=1}^{\bar{G}} \lambda_g\|\beta_{[g]}\|, \tag{31}$$

where $\mathbf{Y}_{n\times 1} = \mathcal{X}_i^T$, $\mathbf{X}_{n\times\bar{p}} = [\mathcal{X}^1 : \cdots : \mathcal{X}^{T-1}]$, $\beta^0_{\bar{p}\times 1} = vec(A_{i:}^{1:(T-1)})$, $\{1, \ldots, \bar{p}\} = \cup_{g=1}^{\bar{G}}\mathcal{G}_g$, $\bar{p} = (T-1)p$, $\bar{G} = (T-1)G$ and $\lambda_g = \lambda w_{i,g}^t$. In Proposition C.1, we first establish the upper bounds on estimation error in the context of a generic group lasso penalized regression problem. The results for regular group NGC then readily follows by applying the above Proposition on the p separate regressions.

Recall the Restricted Eigenvalue assumption required for the derivation of ℓ_2 estimation and prediction error. Following van de Geer and Bühlmann (2009), we introduce a slightly weaker notion called **Group Compatibility** (GC). For a constant $L > 0$ we say that GC(S, L) condition holds, if there exists a constant
$\phi_{compatible} = \phi_{compatible}(S, L) > 0$ such that

$$\min_{\Delta\in\mathbb{R}^p\setminus\{\mathbf{0}\}}\left\{\frac{\left(\sum_{g\in S}\lambda_g^2\right)^{1/2}\|X\Delta\|}{\sqrt{n}\sum_{g\in S}\lambda_g\|\Delta_{[g]}\|} : \sum_{g\notin S}\lambda_g\|\Delta_{[g]}\| \leq L\sum_{g\in S}\lambda_g\|\Delta_{[g]}\|\right\} \geq \phi_{compatible}. \tag{32}$$

The fact that GC(S, L) holds whenever RE(s, L) is satisfied (and $\phi_{RE} \leq \phi_{compatible}$) follows at once from Cauchy Schwarz inequality. We shall derive upper bounds on the prediction and $\ell_{2,1}$ estimation error of group lasso estimates involving the compatibility constant. This notion will also be used later to connect the irrepresentable conditions to the consistency results of group lasso estimators.

Proposition C.1 *Suppose the GC condition* (32) *holds with* $L = 3$. *Choose* $\alpha > 0$ *and denote* $\lambda_{min} = \min_{1 \leq g \leq G} \lambda_g$. *If*

$$\lambda_g \geq \frac{2\sigma}{\sqrt{n}} \sqrt{\left\| C_{[g][g]} \right\|} \left(\sqrt{k_g} + \frac{\pi}{\sqrt{2}} \sqrt{\alpha \log G} \right)$$

for every $g \in \mathbb{N}_G$, *then, the following statements hold with probability at least* $1 - 2G^{1-\alpha}$,

$$\frac{1}{n} \left\| X \left(\hat{\beta} - \beta^0 \right) \right\|^2 \leq \frac{16}{\phi^2_{compatible}} \sum_{g=1}^{s} \lambda_g^2, \tag{33}$$

$$\|\hat{\beta} - \beta^0\|_{2,1} \leq \frac{16}{\phi^2_{compatible}} \frac{\sum_{g=1}^{s} \lambda_g^2}{\lambda_{min}}. \tag{34}$$

If, in addition, RE(2s, 3) holds, then, with the same probability we get

$$\|\hat{\beta} - \beta^0\| \leq \frac{4\sqrt{10}}{\phi^2_{RE}(2s)} \frac{\sum_{g=1}^{s} \lambda_g^2}{\lambda_{min} \sqrt{s}}. \tag{35}$$

Proof [Proof of Proposition (C.1)] Since $\hat{\beta}$ is a solution of the optimization problem (31), for all $\beta \in \mathbb{R}^p$, we have

$$\frac{1}{n} \|Y - X\hat{\beta}\|^2 + 2 \sum_{g=1}^{G} \lambda_g \|\hat{\beta}_{[g]}\| \leq \frac{1}{n} \|Y - X\beta\|^2 + 2 \sum_{g=1}^{G} \lambda_g \|\beta_{[g]}\|.$$

Plugging in $Y = X\beta^0 + \epsilon$, and simplifying the resulting equation, we get

$$\frac{1}{n} \|X(\hat{\beta} - \beta^0)\|^2 \leq \frac{1}{n} \|X(\beta - \beta^0)\|^2 + \frac{2}{n} \sum_{g=1}^{G} \left\| (X'\epsilon)_{[g]} \right\| \left\| (\hat{\beta} - \beta)_{[g]} \right\|$$
$$+ 2 \sum_{g=1}^{G} \lambda_g \left(\|\beta_{[g]}\| - \|\hat{\beta}_{[g]}\| \right).$$

Fix $g \in \mathbb{N}_G$ and consider the event $\mathcal{A}_g = \left\{ \epsilon \in \mathbb{R}^n : \frac{2}{n} \left\| (X'\epsilon)_{[g]} \right\| \leq \lambda_g \right\}$. Note that $Z = \frac{1}{\sqrt{n}} X'\epsilon \sim N(\mathbf{0}, \sigma^2 C)$. So $Z_{[g]} \sim N(\mathbf{0}, \sigma^2 C_{[g][g]})$. Then,

$$\begin{aligned} \mathbb{P}\left(\mathcal{A}_g^c\right) &= \mathbb{P}\left(\left\| Z_{[g]} \right\| > \frac{1}{2} \lambda_g \sqrt{n} \right) \\ &\leq \mathbb{P}\left(\left| \left\| Z_{[g]} \right\| - \mathbb{E} \left\| Z_{[g]} \right\| \right| > \frac{\lambda_g \sqrt{n}}{2} - \sigma \sqrt{k_g} \sqrt{\left\| C_{[g][g]} \right\|} \right), \end{aligned}$$

where the last inequality follows from the second statement of Lemma A.2. Now, let $x_g = \frac{\lambda_g\sqrt{n}}{2} - \sigma\sqrt{k_g}\sqrt{\left\|C_{[g][g]}\right\|}$. Then, for $x_g > 0$, if

$$2\exp\left(-\frac{2\,x_g^2}{\pi^2\sigma^2\left\|C_{[g][g]}\right\|}\right) \leq 2\,G^{-\alpha}\,,$$

we get

$$\mathbb{P}\left(\mathcal{A}_g^c\right) \leq 2G^{-\alpha}.$$

But this happens if,

$$\sqrt{2}x_g \geq \sqrt{\alpha\,\log\,G}\pi\sigma\sqrt{\left\|C_{[g][g]}\right\|},$$

which is ensured by the proposed choice of λ_g.

Next, define $\mathcal{A} := \cap_{g=1}^{G}\mathcal{A}_g$. Then, $\mathbb{P}\left(\mathcal{A}\right) \geq 1 - 2G^{1-\alpha}$, and on the event $\mathcal{A}$, we have, for all $\beta \in \mathbb{R}^p$,

$$\frac{1}{n}\|X(\hat{\beta} - \beta^0)\|^2 + \sum_{g=1}^{G}\lambda_g\left\|\hat{\beta}_{[g]} - \beta_{[g]}\right\| \leq \frac{1}{n}\|X(\beta - \beta^0)\|^2 + 2\sum_{g=1}^{G}\lambda_g\left(\left\|\hat{\beta}_{[g]} - \beta_{[g]}\right\| + \left\|\beta_{[g]}\right\| - \left\|\hat{\beta}_{[g]}\right\|\right).$$

Note that $\left(\left\|\hat{\beta}_{[g]} - \beta_{[g]}\right\| + \left\|\beta_{[g]}\right\| - \left\|\hat{\beta}_{[g]}\right\|\right)$ vanishes if $g \notin S$ and is bounded above by $\min\{2\left\|\beta_{[g]}\right\|, 2\left(\left\|\beta_{[g]} - \hat{\beta}_{[g]}\right\|\right)\}$ if $g \in S$.

This leads to the following sparsity oracle inequality, for all $\beta \in \mathbb{R}^p$,

$$\frac{1}{n}\|X(\hat{\beta} - \beta^0)\|^2 + \sum_{g=1}^{G}\lambda_g\left\|\hat{\beta}_{[g]} - \beta_{[g]}\right\| \leq \frac{1}{n}\|X(\beta - \beta^0)\|^2 + 4\sum_{g\in S}\lambda_g \min\left\{\left\|\beta_{[g]}\right\|, \left\|\beta_{[g]} - \hat{\beta}_{[g]}\right\|\right\}. \tag{36}$$

The sparsity oracle inequality (36) with $\beta = \beta^0$, and $\Delta := \hat{\beta} - \beta^0$ leads to the following two useful bounds on the prediction and $\ell_{2,1}$-estimation errors:

$$\frac{1}{n}\left\|X\Delta\right\|^2 \leq 4\sum_{g\in S}\lambda_g\left\|\Delta_{[g]}\right\|, \tag{37}$$

$$\sum_{g\notin S}\lambda_g\left\|\Delta_{[g]}\right\| \leq 3\sum_{g\in S}\lambda_g\left\|\Delta_{[g]}\right\|. \tag{38}$$

Now, assume the group compatibility condition 32 holds. Then,

$$\frac{1}{n}\left\|X\Delta\right\|^2 \leq 4\sum_{g\in S}\lambda_g\left\|\Delta_{[g]}\right\| \leq \sqrt{\sum_{g\in S}\lambda_g^2}\,\frac{\|X\Delta\|}{\sqrt{n}}\,\frac{4}{\phi_{compatible}}, \tag{39}$$

which implies the first inequality of proposition C.1. The second inequality follows from

$$\begin{aligned}\lambda_{min}\left\|\hat{\beta}-\beta\right\|_{2,1} &\leq \sum_{g=1}^{G}\lambda_g\left\|\Delta_{[g]}\right\| \leq 4\sum_{g\in S}\lambda_g\left\|\Delta_{[g]}\right\| \\ &\leq 4\sqrt{\sum_{g\in S}\lambda_g^2}\,\frac{\|X\Delta\|}{\sqrt{n}}\,\frac{1}{\phi_{compatible}} \leq \frac{16}{\phi^2_{compatible}}\sum_{g\in S}\lambda_g^2,\end{aligned}$$

where the last step uses (39).

The proof of the last inequality of proposition C.1, i.e., the upper bound on ℓ_2 estimation error under $RE(2s)$, is the same as in Theorem 3.1 in Lounici et al. (2011) and is omitted. ■

Proof [Proof of Proposition 3.1] Applying the ℓ_2-estimation error of (35) on the i^{th} group lasso regression problem of regular group NGC, we have

$$\|\hat{A}_{i:}^{1:T-1}-A_{i:}^{1:T-1}\| \leq \frac{4\sqrt{10}}{\phi^2_{RE}(2s_i)}\frac{\sum_{g=1}^{s_i}\lambda_g^2}{\lambda_{\min}\sqrt{s_i}} \leq \frac{4\sqrt{10}}{\phi^2_{RE}(2s_{\max})}\frac{\lambda_{\max}}{\lambda_{\min}}\sqrt{s_i}$$

with probability at least $1-2\bar{G}^{1-\alpha}$. Combining the bounds for all $i=1,\ldots,p$ and noting that $s=\sum_{i=1}^{p}s_i$, we have the required result. ■

Appendix D. Irrepresentable assumptions and consistency

In this section, we discuss two results involving the compatibility and irrepresentable conditions for group lasso. We first show that a stronger version of the uniform irrepresentable assumption implies the group compatibility (32), and hence, consistency in $\ell_{2,1}$ norm. Next we argue that a weaker version of the irrepresentable assumption is indeed necessary for the direction consistency of the group lasso estimates. These results generalize analogous properties of lasso (van de Geer and Bühlmann, 2009; Zhao and Yu, 2006) to the group penalization framework. The proofs are given under a special choice of tuning parameter $\lambda_g=\lambda\sqrt{k_g}$. Similar results can be derived for the general choice of λ_g, although their presentation is more involved.

Proposition D.1 *Assume uniform irrepresentable condition* (13) *holds with* $\eta\in(0,1)$*, and* $\Lambda_{min}(C_{11})>0$*. Then group compatibility(S, L)* (32) *condition holds whenever* $L<\frac{1}{1-\eta}$*.*

Proof First note that with the above choice of λ_g the Group Compatibility (S,L) condition simplifies to

$$\phi_{compatible}:=\min_{\Delta\in\mathbb{R}^p\setminus\{\mathbf{0}\}}\left\{\frac{\sqrt{q}\|X\Delta\|}{\sqrt{n}\sum_{g\in S}\sqrt{k_g}\|\Delta_{[g]}\|}:\sum_{g\notin S}\sqrt{k_g}\|\Delta_{[g]}\|\leq L\sum_{g\in S}\sqrt{k_g}\|\Delta_{[g]}\|\right\}>0. \tag{40}$$

Also, the uniform irrepresentable condition guarantees that there exists $0 < \eta < 1$ such that $\forall \tau \in \mathbb{R}^q$ with $\|\tau\|_{2,\infty} = \max_{1 \le g \le s} \|\tau_{[g]}\|_2 \le 1$, we have,

$$\frac{1}{\sqrt{k_g}} \left\| \left[C_{21} \left(C_{11} \right)^{-1} K^0 \tau \right]_{[g]} \right\|_2 < 1 - \eta \ \forall g \notin S.$$

Here $K^0 = K/\lambda$ is a $q \times q$ block diagonal matrix with diagonal blocks $\sqrt{k_1}\, \mathbf{I}_{k_1 \times k_1}, \ldots, \sqrt{k_s}\, \mathbf{I}_{k_s \times k_s}$. Define

$$\Delta^0 := \underset{\Delta \in \mathbb{R}^p}{\operatorname{argmin}} \left\{ \frac{1}{2n} \|\mathbf{X}\Delta\|_2^2 : \quad \sum_{g \in S} \sqrt{k_g} \|\Delta_{[g]}\|_2 = 1, \ \sum_{g \notin S} \sqrt{k_g} \|\Delta_{[g]}\|_2 \le L \right\}. \tag{41}$$

Note that $\frac{1}{n}\|\mathbf{X}\Delta^0\|_2^2 = \phi^2_{compatible}/q$, and introduce two Lagrange multipliers λ and λ' corresponding to the equality and inequality constraints for solving the optimization problem in (41). Also, partition $\Delta^0 = \left[\Delta^0_{(1)} : \Delta^0_{(2)} \right]$ and $\mathbf{X} = \left[\mathbf{X}_{(1)} : \mathbf{X}_{(2)} \right]$ into signal and nonsignal parts as in (10). The first q linear equations of the KKT conditions imply that there exists $\tau^0 \in \mathbb{R}^q$ such that

$$C_{11}\Delta^0_{(1)} + C_{12}\Delta^0_{(2)} \quad = \quad \lambda K^0 \tau^0 \tag{42}$$

and, for every $g \in S$,

$$\begin{aligned} \tau^0_{[g]} \quad &= \quad D(\Delta^0_{[g]}) \text{ if } \Delta^0_{[g]} \neq \mathbf{0}, \\ \|\tau^0_{[g]}\|_2 \quad &\le \quad 1 \text{ if } \Delta^0_{[g]} = \mathbf{0}. \end{aligned}$$

It readily follows that $\left(\tau^0\right)^T K^0 \Delta^0_{(1)} = \sum_{g \in S} \sqrt{k_g} \|\Delta^0_{[g]}\|_2 = 1$.

Multiplying both sides of (42) by $\left(\Delta^0_{(1)}\right)^T$ we get

$$\left(\Delta^0_{(1)}\right)^T C_{11} \Delta^0_{(1)} + \left(\Delta^0_{(1)}\right)^T C_{12} \Delta^0_{(2)} = \lambda. \tag{43}$$

Also, (42) implies

$$\Delta^0_{(1)} + \left(C_{11}\right)^{-1} C_{12} \Delta^0_{(2)} = \lambda \left(C_{11}\right)^{-1} K^0 \tau^0. \tag{44}$$

Multiplying both sides of the equation by $\left(K^0\tau^0\right)^T = \left(\tau^0\right)^T K^0$ we obtain

$$1 = -\left(\tau^0\right)^T K^0 \left(C_{11}\right)^{-1} C_{12} \Delta^0_{(2)} + \lambda \left(K^0\tau^0\right)^T \left(C_{11}\right)^{-1} \left(K^0\tau^0\right). \tag{45}$$

Note that the absolute value of the first term,

$$\left| \sum_{g \notin S} \left(\Delta^0_{[g]}\right)^T \left[C_{21} (C_{11})^{-1} K^0 \tau^0 \right]_{[g]} \right|, \tag{46}$$

is bounded above by

$$(1-\eta) \left(\sum_{g \notin S} \sqrt{k_g} \|\Delta^0_{[g]}\|_2 \right) \le (1-\eta) L \tag{47}$$

by virtue of the uniform irrepresentable condition and the Cauchy-Schwartz inequality. Assuming the minimum eigenvalue of C_{11}, i.e., $\Lambda_{min}(C_{11})$, is positive and considering $\|K^0\tau^0\|_2 \leq \sqrt{q}$, the second term is at most $\lambda\, q/\Lambda_{min}(C_{11})$. So (45) implies

$$1 \leq (1-\eta)L + \frac{\lambda q}{\Lambda_{min}(C_{11})}. \tag{48}$$

In particular, $\lambda \geq \Lambda_{min}(C_{11})\,(1-(1-\eta)L)\,/q$ is positive whenever $L < 1/(1-\eta)$.
Next, multiply both sides of (44) by $(\Delta^0_{(2)})^T C_{21}$ to get

$$\left(\Delta^0_{(2)}\right)^T C_{21}\Delta^0_{(1)} + \left(\Delta^0_{(2)}\right)^T C_{21}\,(C_{11})^{-1}\,C_{(12)}\Delta^0_{(2)} = \lambda\left(\Delta^0_{(2)}\right)^T C_{21}\,(C_{11})^{-1}\,K^0\tau^0. \tag{49}$$

Using the upper bound in (47), the right hand side is at least $-\lambda(1-\eta)L$.
Also a simple consequence of the block inversion formula of the non-negative definite matrix C guarantees that the matrix $C_{22} - C_{21}\,(C_{11})^{-1}\,C_{12}$ is non-negative definite. Hence,

$$\left(\Delta^0_{(2)}\right)^T\left[C_{22} - C_{21}\,(C_{11})^{-1}\,C_{12}\right]\Delta^0_{(2)} \geq 0$$

$$\text{and} \quad \left(\Delta^0_{(2)}\right)^T C_{22}\Delta^0_{(2)} \geq \left(\Delta^0_{(2)}\right)^T C_{21}\,(C_{11})^{-1}\,C_{12}\Delta^0_{(2)}.$$

Putting all the pieces together we get

$$\begin{aligned}
\phi^2_{compatible}/q &= \frac{1}{n}\|\mathbf{X}\Delta^0\|_2^2 \\
&= {\Delta^0_{(1)}}^T C_{11}\Delta^0_{(1)} + 2{\Delta^0_{(2)}}^T C_{21}\Delta^0_{(1)} + {\Delta^0_{(2)}}^T C_{22}\Delta^0_{(2)} \\
&= \lambda + {\Delta^0_{(2)}}^T C_{21}\Delta^0_{(1)} + {\Delta^0_{(2)}}^T C_{22}\Delta^0_{(2)}\ \text{, by (43)} \\
&\geq \lambda - \lambda(1-\eta)L\ \text{, by (49)} \\
&= \lambda(1-(1-\eta)L).
\end{aligned}$$

Plugging in the lower bound for λ we obtain the result; namely,

$$\phi^2_{compatible} = \Lambda_{min}(C_{11})\,(1-(1-\eta)L)^2 > 0$$

for any $L < \frac{1}{1-\eta}$. ■

In this section we investigate the necessity of irrepresentable assumptions for direction consistency of group lasso estimates. To this end we first introduce the notion of weak irrepresentability.

For a q-dimensional vector τ define the stacked direction vector

$$\underbrace{\tilde{D}(\tau)}_{q\times 1} = [\underbrace{D(\tau_{[1]})'}_{k_1\times 1}, \ldots, \underbrace{D(\tau_{[s]})'}_{k_s\times 1}]'.$$

Weak Irrepresentable Condition is satisfied if

$$\frac{1}{\lambda_g}\left\|\left[C_{21}(C_{11})^{-1}K\tilde{D}(\beta^0_{(1)})\right]_{[g]}\right\| \leq 1,\ \forall g \notin S = \{1,\ldots,s\}. \tag{50}$$

We argue the necessity of weak irrepresentable condition for group sparsity selection and direction consistency under two regularity conditions on the design matrix, as n, $p \to \infty$:
(A1) The minimum eigenvalue of the signal part of the Gram matrix, viz. $\Lambda_{min}(C_{11})$, is bounded away from zero.
(A2) The matrices C_{21} and C_{22} are bounded above in spectral norm.

As in the last proposition, we set $\lambda_g = \lambda\sqrt{k_g}$ and $K^0 = K/\lambda$. Suppose that the weak irrepresentable condition does not hold, i.e., for some $g \notin S$ and $\xi > 0$, we have,

$$\frac{1}{\sqrt{k_g}} \left\| \left[C_{21}(C_{11})^{-1} K^0 \tilde{D}(\beta^0_{(1)}) \right]_{[g]} \right\| > 1 + \xi$$

for infinitely many n. Also suppose that there exists a sequence of positive reals $\delta_n \to 0$ such that the event

$$E_n := \{ \|D(\hat{\beta}_{[g]}) - D(\beta_{[g]})\|_2 < \delta_n, \ \forall g \in S, \text{ and } \hat{\beta}_{[g]} = \mathbf{0} \, \forall \, g \notin S \}$$

satisfies $\mathbb{P}(E_n) \to 1$ as p, $n \to \infty$.

Note that for large enough n so that $\delta_n < \min_g \|D(\beta_{[g]})\|$, we have $\hat{\beta}_{[g]} \neq \mathbf{0}$, $\forall \, g \in S$ on the event E_n.

Then, as in the proof of Theorem 4.1, we have, on the event E_n,

$$\hat{\mathbf{u}} = (C_{11})^{-1} \left[\frac{1}{\sqrt{n}} Z_{(1)} - \lambda K^0 \tilde{D}(\hat{\beta}_{(1)}) \right] \tag{51}$$

$$\text{and} \quad \frac{1}{n} \left\| \left[{\mathbf{X}_{(2)}}^T (\epsilon - \mathbf{X}_{(1)} \hat{u}) \right]_{[g]} \right\| \leq \lambda \sqrt{k_g}, \ \forall g \notin S. \tag{52}$$

Substituting the value of $\hat{u}$ from (51) in (52), we have, on the event E_n,

$$\frac{1}{\sqrt{n}} \left\| \left[\mathbf{Z}_{(2)} - C_{21}(C_{11})^{-1} \mathbf{Z}_{(1)} + \lambda\sqrt{n} C_{21}(C_{11})^{-1} K^0 \tilde{D}(\hat{\beta}_{(1)}) \right]_{[g]} \right\| \leq \lambda\sqrt{k_g},$$

which implies that

$$\begin{aligned} &\left\| \left[Z_{(2)} - C_{21} \left(C_{11}\right)^{-1} Z_{(1)} \right]_{[g]} \right\| \\ &\qquad \geq \lambda \sqrt{n} \sqrt{k_g} \left[\frac{1}{\sqrt{k_g}} \left\| \left[C_{21}(C_{11})^{-1} K^0 \tilde{D}(\hat{\beta}_{(1)}) \right]_{[g]} \right\| - 1 \right]. \end{aligned} \tag{53}$$

Now note that for large enough n, if $\|C_{21}\|$ is bounded above, direction consistency guarantees that the expression on the right is larger than

$$\frac{1}{2} \lambda \sqrt{n} \sqrt{k_g} \left[\frac{1}{\sqrt{k_g}} \left\| \left[C_{21}(C_{11})^{-1} K^0 \tilde{D}(\beta_{(1)}) \right]_{[g]} \right\| - 1 \right],$$

which in turn is larger than $\frac{1}{2} \lambda \sqrt{n} \sqrt{k_g} \, \xi$, in view of the weak irrepresentable condition.

This contradicts $\mathbb{P}(E_n) \to 1$, since the left-hand side of (53) corresponds to the norm of a centered Gaussian random variable with bounded variance structure $\left[C_{22} - C_{21} C_{11}^{-1} C_{12} \right]_{[g][g]}$ while $\lambda \sqrt{n} \sqrt{k_g}$ diverges with $\sqrt{\log G}$.

Appendix E. Thresholding Group Lasso Estimates.

Proof [Proof of Theorem 4.2] We use the notations developed in the proof of Proposition C.1. First note that, (ii) follows directly from Theorem 4.1. For (i), since the falsely selected groups are present after the initial thresholding, we get $\|\hat{\beta}_{[g]}\| > 4\lambda$ for every such group. Next, we obtain an upper bound for the number of such groups. Specifically, denoting $\Delta = \hat{\beta} - \beta^0$, we get

$$\left|\hat{S}\backslash S\right| \leq \frac{\|\hat{\beta}_{S^c}\|_{2,1}}{4\lambda} = \frac{\sum_{g\notin S}\|\Delta_{[g]}\|}{4\lambda}. \tag{54}$$

Next, note that from the sparsity oracle inequality (37), the following holds on the event $\mathcal{A}$,

$$\sum_{g\notin S}\|\Delta_{[g]}\| \leq 3\sum_{g\in S}\|\Delta_{[g]}\|.$$

It readily follows that

$$4\sum_{g\notin S}\|\Delta_{[g]}\| \leq 3\|\Delta\|_{2,1} \leq \frac{48}{\phi^2}\, s\lambda,$$

where the last inequality follows from the $\ell_{2,1}$-error bound of (34). Using this inequality together with (54) gives the result. ■

References

F. R. Bach. Consistency of the group lasso and multiple kernel learning. *J. Mach. Learn. Res.*, 9:1179–1225, 2008. ISSN 1532-4435.

P. J. Bickel, Y. Ritov, and A. B. Tsybakov. Simultaneous analysis of lasso and dantzig selector. *The Annals of Statistics*, 37(4):1705–1732, 2009.

M. Binder, C. Hsiao, and M.H. Pesaran. Estimation and inference in short panel vector autoregressions with unit roots and cointegration. *Econometric Theory*, 21:795–837, 2005. ISSN 1469-4360. doi: 10.1017/S0266466605050413.

O. Blanchard and R. Perotti. An empirical characterization of the dynamic effects of changes in government spending and taxes on output. *The Quarterly Journal of Economics*, 117 (4):1329–1368, 2002.

P. Breheny and J. Huang. Penalized methods for bi-level variable selection. *Stat. Interface*, 2(3):369–380, 2009. ISSN 1938-7989.

B. Cao and Y. Sun. Asymptotic distributions of impulse response functions in short panel vector autoregressions. *Journal of Econometrics*, 163(2):127 – 143, 2011. ISSN 0304-4076.

N. Friedman. Inferring cellular networks using probabilistic graphical models. *Science's STKE*, 303(5659):799, 2004.

A. Fujita, J. Sato, H. Garay-Malpartida, R. Yamaguchi, S. Miyano, M. Sogayar, and C. Ferreira. Modeling gene expression regulatory networks with the sparse vector autoregressive model. *BMC Systems Biology*, 1(1):39, 2007. ISSN 1752-0509.

G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, third edition, 1996. ISBN 0-8018-5413-X; 0-8018-5414-8.

C. W. J. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3):424–438, 1969.

C. Hiemstra and J. D. Jones. Testing for linear and nonlinear granger causality in the stock price-volume relation. *Journal of Finance*, pages 1639–1664, 1994.

J. Huang and T. Zhang. The benefit of group sparsity. *Ann. Statist.*, 38(4):1978–2004, 2010. ISSN 0090-5364.

J. Huang, S. Ma, H. Xie, and C-H. Zhang. A group bridge approach for variable selection. *Biometrika*, 96(2):339–355, 2009. ISSN 0006-3444.

K. Kim, J.H. Kim, J. Lee, H.M. Jin, S.H. Lee, D.E. Fisher, H. Kook, K.K. Kim, Y. Choi, and N. Kim. Nuclear factor of activated t cells c1 induces osteoclast-associated receptor gene expression during tumor necrosis factor-related activation-induced cytokine-mediated osteoclastogenesis. *Journal of Biological Chemistry*, 280(42):35209–35216, 2005.

M. Ledoux and M. Talagrand. *Probability in Banach Spaces*, volume 23 of *Ergebnisse der Mathematik und ihrer Grenzgebiete (3) [Results in Mathematics and Related Areas (3)]*. Springer-Verlag, Berlin, 1991. ISBN 3-540-52013-9. Isoperimetry and processes.

K. Lounici, M. Pontil, S. van de Geer, and A. B. Tsybakov. Oracle inequalities and optimal inference under group sparsity. *Ann. Statist.*, 39(4):2164–2204, 2011.

A. Lozano, N. Abe, Y. Liu, and S. Rosset. Grouped graphical granger modeling for gene expression regulatory networks discovery. *Bioinformatics*, 25(12):i110, 2009.

H. Lütkepohl. *New Introduction to Multiple Time Series Analysis*. Springer, 2005.

G. Michailidis. Statistical challenges in biological networks. *Journal of Computational and Graphical Statistics*, 21(4):840–855, 2012. doi: 10.1080/10618600.2012.738614.

S. V. Parter. Extreme eigenvalues of Toeplitz forms and applications to elliptic difference equations. *Trans. Amer. Math. Soc.*, 99:153–192, 1961. ISSN 0002-9947.

J. Pearl. *Causality: Models, Reasoning, and Inference*, volume 47. Cambridge, 2000.

M. B. Priestley. *Spectral Analysis and Time Series. Vol. 2*. Academic Press Inc. [Harcourt Brace Jovanovich Publishers], London, 1981. ISBN 0-12-564902-9. Multivariate series, prediction and control, Probability and Mathematical Statistics.

C. Rangel, J. Angus, Z. Ghahramani, M. Lioumi, E. Sotheran, A. Gaiba, D.L. Wild, and F. Falciani. Modeling t-cell activation using gene expression profiling and state-space models. *Bioinformatics*, 20(9):1361, 2004.

M. Rudelson and R. Vershynin. Hanson-wright inequality and sub-gaussian concentration. *Electron. Commun. Probab.*, 18:no. 82, 1–9, 2013. ISSN 1083-589X. doi: 10.1214/ECP.v18-2865.

A. Shojaie and G. Michailidis. Penalized likelihood methods for estimation of sparse high dimensional directed acyclic graphs. *Biometrika*, 97(3):519–538, 2010a.

A. Shojaie and G. Michailidis. Discovering graphical granger causality using a truncating lasso penalty. *Bioinformatics*, 26(18):i517–i523, 2010b.

C.A. Sims. Money, income, and causality. *The American Economic Review*, 62(4):540–552, 1972.

S. van de Geer and P. Bühlmann. On the conditions used to prove oracle results for the Lasso. *Electron. J. Stat.*, 3:1360–1392, 2009. ISSN 1935-7524.

S. van de Geer, P. Bühlmann, and S. Zhou. The adaptive and the thresholded Lasso for potentially misspecified models (and a lower bound for the Lasso). *Electron. J. Stat.*, 5: 688–749, 2011. ISSN 1935-7524.

R. Vershynin. *Lectures in Geometric Functional Analysis*. available at http://www-personal.umich.edu/ romanv/papers/GFA-book/GFA-book.pdf, 2009.

M.J. Wainwright. Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_1-constrained quadratic programming (Lasso). *IEEE Transactions on Information Theory*, 55(5):2183 –2202, May 2009. ISSN 0018-9448.

M. L. Waterman, K. A. Jones, et al. Purification of tcf-1 alpha, a t-cell-specific transcription factor that activates the t-cell receptor c alpha gene enhancer in a context-dependent manner. *The New Biologist*, 2(7):621, 1990.

F. Wei and J. Huang. Consistent group selection in high-dimensional linear regression. *Bernoulli*, 16(4):1369–1384, 2010. ISSN 1350-7265.

P. Zhao and B. Yu. On model selection consistency of lasso. *J. Mach. Learn. Res.*, 7: 2541–2563, December 2006. ISSN 1532-4435.

S. Zhou. Thresholded lasso for high dimensional variable selection and statistical estimation. *Arxiv preprint arXiv:1002.1583*, 2010.

Journal of Machine Learning Research 16 (2015) 455-490 Submitted 3/14; Revised 8/14; Published 3/15

Iterative and Active Graph Clustering Using Trace Norm Minimization Without Cluster Size Constraints*

Nir Ailon NAILON@CS.TECHNION.AC.IL
Department of Computer Science
Technion IIT Haifa, Israel

Yudong Chen YUDONG.CHEN@EECS.BERKELEY.EDU
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley
Berkeley, CA 94720, USA

Huan Xu MPEXUH@NUS.EDU.SG
Department of Mechanical Engineering
National University of Singapore
Singapore 117575

Editor: Tong Zhang

Abstract

This paper investigates graph clustering under the planted partition model in the presence of *small clusters*. Traditional results dictate that for an algorithm to provably correctly recover the underlying clusters, all clusters must be sufficiently large—in particular, the cluster sizes need to be $\tilde{\Omega}(\sqrt{n})$, where n is the number of nodes of the graph. We show that this is not really a restriction: by a refined analysis of a convex-optimization-based recovery approach, we prove that small clusters, under certain mild assumptions, do not hinder recovery of large ones. Based on this result, we further devise an iterative algorithm to provably recover *almost all clusters* via a "peeling strategy": we recover large clusters first, leading to a reduced problem, and repeat this procedure. These results are extended to the partial observation setting, in which only a (chosen) part of the graph is observed. The peeling strategy gives rise to an *active* learning algorithm, in which edges adjacent to smaller clusters are queried more often after large clusters are learned (and removed). We expect that the idea of iterative peeling—that is, sequentially identifying a subset of the clusters and reducing the problem to a smaller one—is useful more broadly beyond the specific implementations (based on convex optimization) used in this paper.

Keywords: graph clustering, community detection, active clustering, convex optimization, planted partition model, stochastic block model

1. Introduction

This paper considers the following classic graph clustering problem: given an undirected unweighted graph, partition the nodes into disjoint clusters so that the density of edges within each cluster is higher than those across clusters. Graph clustering arises naturally in many applications across science and engineering; prominent examples include community

*. This work extends and improves a preliminary conference version Ailon et al. (2013).

detection in social networks (Mishra et al., 2007; Zhao et al., 2011), submarket identification in E-commerce and sponsored search (Yahoo!-Inc, 2009), and co-authorship analysis in document database (Ester et al., 1995), among others. From a purely binary classification theoretical point of view, the edges of the graph are (noisy) labels of "similarity" or "affinity" between pairs of objects, and the concept class consists of clusterings of the objects (encoded graphically by identifying clusters with cliques).

Many theoretical results in graph clustering consider the *Planted Partition Model* (Condon and Karp, 2001), in which the edges are generated randomly based on an unknown set of underlying clusters; see Section 1.1 for more details. While numerous different methods have been proposed, their performance guarantees under the planted partition model generally have the following form: under certain conditions of the density of edges (within clusters and across clusters), the method succeeds to recover the correct clusters exactly *if all clusters are larger than a threshold size*, typically $\tilde{\Omega}(\sqrt{n})$;[1] see e.g., McSherry (2001); Bollobás and Scott (2004); Ames and Vavasis (2011); Chen et al. (2012); Chaudhuri et al. (2012); Anandkumar et al. (2014).

In this paper, we aim to relax this cluster size constraint of graph clustering under the planted partition model. Identifying extremely small clusters is inherently hard as they are easily confused with "fake" clusters generated by noisy edges,[2] and is not the focus of this paper. Instead, in this paper we investigate a question that has not been addressed before: Can we still recover large clusters in the presence of small clusters? Intuitively, this should be doable. To illustrate, consider an extreme example where the given graph G consists of two subgraphs G_1 and G_2 with disjoint node sets. Suppose G_1, if presented alone, can be correctly clustered using some existing methods, G_2 is a very small clique, and there are relatively few edges connecting G_1 and G_2. The graph G certainly violates the minimum cluster size requirement of previous results, but why should G_2 spoil our ability to correctly cluster G_1?

Our main result confirms this intuition. We show that the cluster size barrier arising in previous work is not really a restriction, but rather an artifact of the attempt to solve the problem in a single shot and recover large and small clusters simultaneously. Using a more careful analysis, we prove that a mixed trace-norm and ℓ_1-norm based convex formulation can recover clusters of size $\tilde{\Omega}(\sqrt{n})$ even in the presence of smaller clusters. That is, small clusters do not interfere with recovery of the large clusters.

The main implication of this result is that one can apply an *iterative* "peeling" strategy to recover smaller and smaller clusters. The intuition is simple: suppose the *number* of clusters is limited, then either all clusters are large, or the sizes of the clusters vary significantly. The first case is obviously easy. But the second is also tractable, for a different reason: using the aforementioned convex formulation, the larger clusters can be correctly identified; if we remove all nodes from these larger clusters, the remaining subgraph contains significantly fewer nodes than the original graph, which leads to a much lower threshold on the size of the cluster for correct recovery, making it possible for correctly identify some

1. The notations $\tilde{\Omega}(\cdot)$ and $\tilde{O}(\cdot)$ ignore logarithmic factors.
2. Indeed, even in a more lenient setup where one clique (i.e., a perfect cluster) of size K is embedded in an Erdos-Renyi graph of n nodes and 0.5 probability of forming an edge, the best known polynomial-time method requires $K = \Omega(\sqrt{n})$ in order to recover the hidden clique, and it has been a long standing open problem to relax this requirement.

smaller clusters. By repeating this procedure, indeed, we can recover the cluster structure for almost all nodes *with no lower bound on the minimal cluster size*. Below we summarize our main contributions and techniques:

1. We provide a refined analysis (Theorem 2) of the mixed trace-norm and ℓ_1-norm convex relaxation approach for exact cluster recovery proposed in Chen et al. (2014a, 2012), focusing on the case where small clusters exist. We show that in the planted partition setting, if each cluster is either large (more precisely, of size at least $\sigma \approx \sqrt{n}$) or small (of size at most σ/C for some global constant $C > 1$), then with high probability, this convex relaxation approach correctly identifies all large clusters while "ignoring" the small ones. In fact, it is possible to arbitrarily increase the tuning parameter σ in quest of an interval $(\sigma/C, \sigma)$ that is disjoint from the set of cluster sizes. The analysis is done by identifying a certain feasible solution to the convex program and proving its almost sure optimality. This solution easily identifies the large clusters. Previous analysis is performed only in the case where all clusters are of size greater than $\sqrt{n}$.

2. We provide a converse (Theorem 5) of the result just described. More precisely, we show that if for some value of the tuning parameter σ, an optimal solution to the convex relaxation program is an exact representation of a collection of large clusters (a partial clustering), then these clusters are actual ground truth clusters, even if the particular interval corresponding to σ isn't really free of cluster sizes. This allows the practitioner to be certain that the optimal solution is useful. Moreover, this has important algorithmic implications for an iterative recovery procedure which we describe below.

3. The last two points imply that if some interval of the form $(\sigma/C, \sigma)$ is free of cluster sizes, then an exhaustive search of this interval will constructively find large clusters, though not necessarily for that particular interval (Theorem 6). Removing the recovered large clusters leads to a reduced problem with a smaller graph. Repeating this procedure gives rise to an iterative algorithm (Algorithm 2), using a "peeling strategy", to recover smaller and smaller clusters that are otherwise impossible to recover. Using this iterative algorithm, we prove that as long as the *number* of clusters is bounded by $O(\log n)$, regardless of the cluster sizes, we can correctly recover the cluster structure for an overwhelming fraction of nodes (Theorem 7). To the best of our knowledge, this is the first result of provably correct graph clustering assuming only an upper bound on the number of clusters, but otherwise no assumption on the cluster sizes.

4. We extend the result to the partial observation setting, where only a fraction of similarity labels (i.e., edge/no edge) are queried. As expected, large clusters can be identified using small observation rates, and a higher rate is needed to find smaller clusters. Hence, the observation rate serves as the tuning parameter. This gives rise to an *active learning algorithm* (Algorithm 4) based on adaptively increasing the rate of sampling in order to hit an interval free of cluster sizes, and spending more queries on smaller subgraphs after we identify large clusters and peel them off. Performance

guarantees are given for this algorithm (Corollary 8–Theorem 11). This active learning scheme requires significantly fewer samples than uniform sampling .

Beside these technical contributions, this paper suggests a new strategy that is potentially useful for general low-rank matrix recovery and other high-dimensional statistical problems, where the data are typically assumed to have certain low-dimensional structures. Many methods have been developed to exploit this *a priori* structural information so that consistent estimation is possible even when the dimensionality of the problem is larger than the number of samples. Our result shows that one may combine these methods with a "peeling strategy" to further push the envelope of learning structured data: by iteratively recovering the easier structural components and reducing the problem complexity, it may be possible to learn complicated structures that are otherwise difficult to recover using existing one-shot approaches.

1.1 Related Work

The literature of graph clustering is too vast for a detailed survey here; we concentrate on the most related work, and in particular those provide provable guarantees on exact cluster recovery.

1.1.1 Planted Partition Model

Also known as the *stochastic block model* (Holland et al., 1983; Condon and Karp, 2001), this classical model assumes that n nodes are partitioned into subsets, referred to as the "true clusters", and a graph is randomly generated as follows: for each pair of nodes, depending on whether or not they belong to the same subset, an edge connecting them is generated with a probability p or q respectively. The goal is to correctly recover the clusters given the random graph. The planted partition model has a large body of literature. Earlier work focused on the setting where the minimal cluster size is $\Theta(n)$ (Boppana, 1987; Condon and Karp, 2001; Carson and Impagliazzo, 2001; Bollobás and Scott, 2004). Subsequently, a number of methods have been proposed methods to handle sublinear cluster sizes, including randomized algorithms (Shamir and Tsur, 2007), spectral clustering (McSherry, 2001; Chaudhuri et al., 2012; Rohe et al., 2011; Kumar and Kannan, 2010), convex optimization based approaches (Jalali et al., 2011; Chen et al., 2014a, 2012; Ames and Vavasis, 2011; Oymak and Hassibi, 2011) and tensor decomposition methods (Anandkumar et al., 2014). See Chen et al. (2014b) for a survey of existing theoretical guarantees for the planted partition model. While the methodology differs, all the work above requires, sometimes implicitly, a constraint on the minimum size of the true clusters; in particular, the size must be $\Omega(\sqrt{n})$. Our analysis is carried under the planted partition model, and our approach requires no constraint on the cluster sizes. We also mention the work of Zhao et al. (2011) for community detection in social networks, which works under a type of planted partition model. Like ours, their algorithm extracts clusters in an iterative manner and is also amenable to outliers. However, their theoretical guarantees are only shown to hold when $n \to \infty$ and the cluster sizes grow linearly with n.

1.1.2 Low-rank and Sparse Matrix Decomposition via Trace Norm

Motivated by robustifying principal component analysis (PCA), several authors (Chandrasekaran et al., 2011; Candès et al., 2011) show that it is possible to recover a low-rank matrix from sparse errors of arbitrary magnitude, where the key ingredient is using the trace norm (also known as the nuclear norm) as a convex surrogate of the rank. Similar results are obtained when the low rank matrix is corrupted by other types of noise (Xu et al., 2012). Of particular relevance to this paper is the work by Jalali et al. (2011), Oymak and Hassibi (2011) and Chen et al. (2012, 2014a), where they apply this approach to graph clustering, and specifically to the planted partition model. These works require the $\widetilde{\Omega}(\sqrt{n})$ bound on the minimal cluster size. Our approach uses the trace norm relaxation, combined with a more refined analysis and an iterative/active peeling strategy.

1.1.3 Active Learning/Active Clustering

Another line of work that motivates this paper is the study of active learning (a setting in which labeled instances are chosen by the learner, rather than by nature), and in particular active learning algorithms for clustering. The most related work is Ailon et al. (2014), who investigated active learning for the *correlation clustering* problem (Bansal et al., 2004), where the goal is to find a set of clusters whose Hamming distance from the graph is minimized. Ailon et al. (2014) obtain a $(1+\varepsilon)$-approximate solution with respect to the optimum, while (actively) querying no more than $O(n\,\mathrm{poly}(\log n, k, \varepsilon^{-1}))$ edges, where k is the number of clusters. Their result imposed no restriction on cluster sizes and hence inspired this work, but differs in at least two major ways. First, Ailon et al. (2014) did not consider *exact* cluster recovery as we do. Second, their guarantees fall in the Empirical Risk Minimization (ERM) framework, with *no* running time guarantees. Our work uses a convex relaxation algorithm, and is hence computationally efficient. The problem of active learning has also been investigated in other setups including clustering based on distance matrix (Voevodski et al., 2012; Shamir and Tishby, 2011), hierarchical clustering (Eriksson et al., 2011; Krishnamurthy et al., 2012) and low-rank matrix/tensor recovery (Krishnamurthy and Singh, 2013). These setups differ significantly from ours..

Remark 1 (A note on a preliminary version of this paper) *The authors published a weaker version of the results in this paper in a preliminary conference paper (Ailon et al., 2013). An exact comparison is stated after each theorem in the text.*

2. Notation and Setup

In this paper the following notations are used. We use $X(i,j)$ to denote the (i,j)-the entry of a matrix X. For a matrix $X \in \mathbb{R}^{n\times n}$ and a subset $S \subseteq [n]$ of size m, the matrix $X[S] \in \mathbb{R}^{m\times m}$ is the principal minor of X corresponding to the set of indexes S. For a matrix M, $\mathrm{s}(M)$ denotes the support of M, namely, the set of index pairs (i,j) such that $M(i,j) \neq 0$. For any subset Φ of $[n]\times[n]$, $\mathcal{P}_\Phi M$ is the matrix that satisfies

$$(\mathcal{P}_\Phi M)(i,j) = \begin{cases} M(i,j), & (i,j) \in \Phi \\ 0, & \text{otherwise.} \end{cases}$$

We now describe the problem setup. Throughout the paper, V denotes a ground set of elements, which we identify with the set $[n] = \{1, \ldots, n\}$. We assume a ground truth clustering of V given by a pairwise disjoint covering $V_1, \ldots, V_k$, where k is the number of clusters. We say $i \sim j$ if $i, j \in V_a$ for some $a \in [k]$, otherwise $i \not\sim j$. We let $n_a := |V_a|$ be the size of the a-th cluster for each $a \in [k]$. For each $i \in [n]$, $\langle i \rangle$ is index of the cluster that contains i, the unique index satisfying $i \in V_{\langle i \rangle}$.

The ground truth clustering matrix, denoted as K^*, is defined as the $n \times n$ matrix so that $K^*(i, j) = 1$ if $i \sim j$, otherwise 0. This is a block diagonal matrix, each block consisting of 1's only, and its rank is k. The input is a symmetric $n \times n$ matrix A, which is a noisy version of K^*. It is generated according to the *planted partition model* with parameters p and q as follows.

> We think of A as the adjacency matrix of an undirected random graph, where the edge (i, j) is in the graph for $i > j$ with probability p_{ij} if $i \sim j$, otherwise with probability q_{ij}, independent of other choices, where we only assume the edge probabilities satisfy $(\min p_{ij}) =: p > q := (\max q_{ij})$.

We use the convention that the diagonal entries of A are all 1. The matrix $B^* := A - K^*$ can be viewed as the noise matrix. Given A, the task is to find the ground truth clusters.

We remark that the setup above is more flexible than the standard planted partition model: we allow the clusters to have different sizes, and the edges probabilities (p_{ij} and q_{ij}) need not be uniform across node pairs (i, j). One consequence is that the node degrees may not be uniform or correlated with the sizes of the associated clusters. Non-uniformity makes some simple heuristics, such as degree counting and single linkage clustering, vulnerable. For example, we cannot distinguish between large and small clusters simply by looking at the node degrees, since nodes in a small cluster may also have high expected degrees. The single linkage clustering approach also fails in the presence of non-uniformity. We illustrate this with an example. Suppose there are $\sqrt{n}$ clusters of equal size, $p = 1$ and $q = 0.1$. We use the number of common neighbors as the distance function in single linkage clustering. If all q_{ij} are equal to q, then it is easy to see that single linkage clustering will succeed, since with high probability node pairs in the same cluster will have more common neighbors than those in different clusters. Yet, this is not true for non-uniform q_{ij}'s. Consider three nodes 1, 2 and 3, where nodes 1 and 2 are in the same cluster, and node 3 belongs to a different cluster. Suppose for all $i > 3$, $q_{1i} = 0$, $q_{2i} = q_{3i} = 0.1$. The expected number of common neighbors between nodes 1 and 2 is $\sqrt{n}$, whereas the expected number of common neighbors between nodes 2 and 3 is $0.2\sqrt{n} + 0.01(n - 2\sqrt{n})$, which is larger than $\sqrt{n}$ for large n and hence single linkage clustering fails. In contrast, the proposed convex-optimization based method can handle such non-uniform settings, as we show in what follows.

3. Main Results

We remind the reader that the trace norm of a matrix is the sum of its singular values, and the (entry-wise) ℓ_1 norm of a matrix M is $\|M\|_1 := \sum_{i,j} |M(i, j)|$. Consider the following convex program, combining the trace norm of a matrix variable K with the ℓ_1 norm of

another matrix variable B using two parameters c_1, c_2 that will be determined later:

$$\text{(CP)} \qquad \min_{K,B\in\mathbb{R}^{n\times n}} \quad \|K\|_* + c_1 \left\|\mathcal{P}_{\mathrm{s}(A)}B\right\|_1 + c_2 \left\|\mathcal{P}_{\mathrm{s}(A)^c}B\right\|_1$$
$$\text{s.t.} \quad K + B = A,$$
$$0 \le K_{ij} \le 1, \forall (i,j).$$

Here the trace norm term in the objective promotes low-rank solutions and thus encourages the matrix K to have the zero-one block-diagonal structure of a clustering matrix. The matrix $\mathcal{P}_{\mathrm{s}(A)}B = \mathcal{P}_{\mathrm{s}(A)}(A-K)$ is non-zero only on the pairs (i,j) between which there is an edge in the graph ($A_{ij}=1$) but the candidate solution has $K_{ij}=0$, and thus $\mathcal{P}_{\mathrm{s}(A)}B$ corresponds to the "cross-cluster disagreements" between A and K. Similarly, the matrix $\mathcal{P}_{\mathrm{s}(A)^c}B$ corresponds to the "in-cluster disagreements". Hence, the last two terms in the objective is the weighted sum of these two types of disagreements. The formulation (CP) can therefore be considered as a convex relaxation of the so-called *weighted correlation clustering* approach (Bansal et al., 2004), whose objective is to find a clustering that minimizes the weighted disagreements. See Oymak and Hassibi (2011); Mathieu and Schudy (2010); Chen et al. (2014a) for related formulations.

Important to subsequent development is the following new theoretical guarantee for the formulation (CP). We show that (CP) identifies the large clusters whose sizes are above a threshold (chosen by the user) even when small clusters are present. The proof is given in Section 5.1.

Theorem 2 *There exist universal constants $b_3 > 1 > b_4 > 0$ such that the following is true. For any (user-specified) parameters $\kappa \ge 1$ and $t \in [\frac{1}{4}p + \frac{3}{4}q, \frac{3}{4}p + \frac{1}{4}q]$, define*

$$\ell_\sharp := b_3 \frac{\kappa\sqrt{p(1-q)n}}{p-q} \max\left\{1, \frac{\sqrt{p(1-q)}\log^4 n}{\kappa(p-q)\sqrt{n}}\right\}, \quad \ell_\flat := b_4 \frac{\kappa\sqrt{p(1-q)n}}{p-q}\ , \tag{1}$$

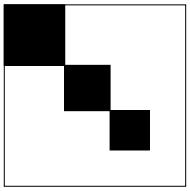
and set

$$c_1 := \frac{1}{100\kappa\sqrt{n}}\sqrt{\frac{1-t}{t}}, \quad c_2 := \frac{1}{100\kappa\sqrt{n}}\sqrt{\frac{t}{1-t}}\ . \tag{2}$$

If (i) $n \ge \ell_\sharp$ and $n \ge 700$, and (ii) for each $a \in [k]$, either $n_a \ge \ell_\sharp$ or $n_a \le \ell_\flat$, then with probability at least $1 - n^{-3}$, the optimal solution to (CP) with c_1, c_2 given above is unique and equal to $(\hat{K}, \hat{B}) = (\mathcal{P}_\sharp K^*, A - \hat{K})$, where for a matrix M, $\mathcal{P}_\sharp M$ is the matrix defined by

$$(\mathcal{P}_\sharp M)(i,j) = \begin{cases} M(i,j), & \max\{n_{\langle i\rangle}, n_{\langle j\rangle}\} \ge \ell_\sharp \\ 0, & \textit{otherwise.} \end{cases}$$

The theorem improves on a weaker version in Ailon et al. 2013, where the ratio $\ell_\sharp/\ell_\flat$ was larger by a factor of $\log^2 n$ than here. The theorem says that the solution to (CP) identifies clusters of size larger than $\ell_\sharp = \Omega(\kappa\sqrt{n})$ and ignores other clusters smaller than $\ell_\flat$. Setting $\kappa = 1$ we recover the usual $\sqrt{n}$ scaling in previous theoretical results. The main novelty here is the treatment of small clusters, whereas in previous work only large clusters were allowed, and there was no guarantee for recovery when small clusters are present.



Black represents 1, white represents 0. Here $\sigma_{\min}(K)$ is the side length of the smallest black square.

Figure 1: Illustration of a partial clustering matrix K.

Note that by the theorem's premise, $\hat{K}$ is the matrix obtained from K^* after zeroing out blocks corresponding to clusters of size at most $\ell_\flat$. Also note that under the assumption

$$p - q \geq \sqrt{p(1-q)} \log^4 n / \sqrt{n} \ , \tag{3}$$

we get the following simpler expression for $\ell_\sharp$ in the theorem, replacing its definition in (1):

$$\ell_\sharp = b_3 \frac{\kappa \sqrt{p(1-q)n}}{p-q} \ . \tag{4}$$

In this case, $\ell_\sharp$ and $\ell_\flat$ differ by only a multiplicative absolute constant b_3/b_4. We will make the assumption (3) in what follows for simplicity, although it is not generally necessary.

Remark 3 *The requirement of having a multiplicative constant gap b_3/b_4 between the sizes $\ell_\sharp$ and $\ell_\flat$ of the large and small clusters, is not an artifact of our analysis; cf. the discussion at the end of Section 4.*

For the convenience of subsequent discussion, we use the following definition.

Definition 4 (Partial Clustering Matrix) *An $n \times n$ matrix K is said to be a partial clustering matrix if there exists a collection of pairwise disjoint sets $U_1, \ldots, U_r \subseteq V$ (called the* induced clusters*) such that $K(i,j) = 1$ if and only if $i, j \in U_a$ for some $a \in [r]$, otherwise 0. If K is a partial clustering matrix then $\sigma_{\min}(K)$ is defined as $\min_{a \in [r]} |U_a|$.*

The definition is depicted in Figure 1. The key message in Theorem 2 is that by choosing κ properly such that no cluster size falls in the interval $(\ell_\flat, \ell_\sharp)$, the unique optimal solution $(\hat{K}, \hat{B})$ to the convex program (CP) is such that $\hat{K}$ is a partial clustering corresponding to large ground truth clusters.

But how can we choose a proper κ? Moreover, given that we chose a κ (say, by exhaustive search), how can we certify that it was indeed chosen properly? In order to develop an algorithm, we would need a type of converse of Theorem 2: There exists an event with high probability (in the random process generating the input graph), such that conditioned on this event, for all values of κ, if an optimal solution to the corresponding (CP) is a partial clustering matrix with the structure illustrated in Figure 1, then the blocks of $\hat{K}$ correspond to ground truth clusters.

Theorem 5 *There exist absolute constants $C_1, C_2 > 0$ such that with probability at least $1 - n^{-3}$, the following holds. For all $\kappa \geq 1$ and $t \in [\frac{3}{4}q + \frac{1}{4}p, \frac{1}{4}q + \frac{3}{4}p]$, if (K, B) is*

an optimal solution to (CP) with c_1, c_2 as defined in Theorem 2, and additionally K is a partial clustering corresponding to $U_1, \ldots, U_r \subseteq V$, with

$$\sigma_{\min}(K) \geq \max\left\{ \frac{C_1 k \log n}{(p-q)^2}, \frac{C_2 \kappa \sqrt{p(1-q)n \log n}}{p-q} \right\}, \tag{5}$$

then $U_1, \ldots, U_r$ are actual ground truth clusters, namely, there exists an injection $\phi : [r] \mapsto [k]$ such that $U_a = V_{\phi(a)}$ for all $a \in [r]$.

Algorithm 1 RecoverBigFullObs(V, A, p, q)

require: ground set V, graph $A \in \mathbb{R}^{V \times V}$, probabilities p, q
$n \leftarrow |V|$
$t \leftarrow \frac{1}{4}p + \frac{3}{4}q$ (or anything in $[\frac{1}{4}p + \frac{3}{4}q, \frac{3}{4}p + \frac{1}{4}q]$)
$\ell_\sharp \leftarrow n$, $g \leftarrow \frac{b_3}{b_4}$
// (If have prior bound k_0 on the number of clusters, take $\ell_\sharp \leftarrow n/k_0$)
while $\ell_\sharp \geq \max\left\{ \frac{C_1 k \log n}{(p-q)^2}, \frac{C_2\sqrt{p(1-q)n \log n}}{p-q} \right\}$ **do**
  solve for κ using (1), set c_1, c_2 as in (2)
  $(K, B) \leftarrow$ optimal solution to (CP) with c_1, c_2
  if K is a partial clustering matrix with $\sigma_{\min}(K) \geq \ell_\sharp$ **then**
    return induced clusters $\{U_1, \ldots, U_r\}$ of K
  end if
  $\ell_\sharp \leftarrow \ell_\sharp / g$
end while
return $\emptyset$

The proof is given in Section 5.2. The combination of Theorems 2 and 5 implies the following, which we state in rough terms for simplicity. Let $g := b_3/b_4$. Assume that we iteratively solve (CP) for κ taking values in some decreasing geometric progression of common ratio g (starting at roughly $\kappa = \sqrt{n}$), and halt if the optimal solution is a partial clustering with clusters of size at least $\ell_\sharp = \ell_\sharp(\kappa)$ (see Algorithm 1). Then these clusters are (extremely likely to be) ground truth clusters. Moreover, if for some κ in the sequence, (i) the interval $(\ell_\flat = \ell_\flat(\kappa), \ell_\sharp = \ell_\sharp(\kappa))$ intersects no cluster size, and (ii) there is at least one cluster at least of size $\ell_\sharp$, then such a halt will (be extremely likely to) occur.

The next question is, when are (i) and (ii) guaranteed? If the number of clusters k is a priori bounded by some k_0, then there is at least one cluster of size at least n/k_0 (alluding to (ii)), and by the pigeonhole principle, any set of $k_0 + 1$ pairwise disjoint intervals of the form $(\alpha, g\alpha)$ contains at least one interval that intersects no clusters size (alluding to (i)). For simplicity, we make an exact quantification of this principle for the case in which p, q are assumed to be fixed and independent of n.[3] As the following theorem shows, it turns out that in this regime, k_0 can be assumed to be asymptotically logarithmic in n to ensure recovery of at least one cluster.[4] In what follows, notation such as $C(p, q), C_3(p, q)$ denotes positive functions that depend on p, q only.

3. In fact, we need only fix $(p - q)$, but we wish to keep this exposition simple.
4. In comparison, Ailon et al. (2014) require k_0 to be constant for their guarantees, as do the Correlation Clustering PTAS in Giotis and Guruswami (2006).

Algorithm 2 RecoverFullObs(V, A, p, q)

require: ground set V, matrix $A \in \mathbb{R}^{V \times V}$, probabilities p, q
$\{U_1, \ldots, U_r\} \leftarrow$ RecoverBigFullObs(V, A, p, q)
$V' \leftarrow [n] \setminus (U_1 \cup \cdots \cup U_r)$
if $r = 0$ **then**
  return $\emptyset$
else
  return RecoverFullObs$(V', A[V'], p, q) \cup \{U_1, \ldots, U_r\}$
end if

Theorem 6 *There exist $C_3(p,q), C_4(p,q), C_5 > 0$ such that the following holds. Assume that $n > C_4(p,q)$, and that we are guaranteed that $k \le k_0$, where $k_0 = C_3(p,q)\log n$. Then with probability at least $1 - 2n^{-3}$, Algorithm 1 will recover at least one cluster in at most $C_5 k_0$ iterations.*

The theorem improves on a counterpart in the preliminary paper (Ailon et al., 2013), where k_0 was smaller by a factor of $\log\log n$ than here.

Proof Consider the set of intervals

$$\left(n/(gk_0), n/k_0\right), \left(n/(g^2k_0), n/(gk_0)\right), \ldots, \left(n/(g^{k_0+1}k_0), n/(g^{k_0}k_0)\right) .$$

By the pigeonhole principle, one of these intervals must not intersect the set of cluster sizes. Assume this interval is $(n/(g^{i_0+1}k_0), n/(g^{i_0}k_0))$, for some $0 \le i_0 \le k_0$. By setting $C_3(p,q)$ small enough so that n/k_0 is at least $\Omega(\sqrt{n\log n})$, and $C_4(p,q)$ large enough so that $n/g^{k_0+1}k_0$ is at least $\Omega(\sqrt{n\log n})$, one easily checks that both the requirements of Theorems 2 and 5 are fulfilled. ■

Theorem 6 ensures that by trying at most a logarithmic number of values of κ, we can recover at least one large cluster, assuming the number of clusters is logarithmic in n. After recovering and removing such a cluster, we are left with an input of size $n' < n$, together with an updated upper bound $k_0' < k_0$ on the number of clusters. As long as k_0' is logarithmic in n', we can continue identifying another large cluster (with respect to the smaller problem) using the same procedure. Clearly, as long as the input size is of size at most $\exp\{C_3(p,q)k_0\}$, we can iteratively continue this process. The following has been proved:

Theorem 7 *Assume an upper bound k_0 on the number k of clusters, and also that n, k_0 satisfy the requirements of Theorem 6. Then with probability at least $1 - 2n^{-2}$, Algorithm 2 recovers clusters covering all but at most $\max\{\exp\{C_3(p,q)k_0\}, C_4(p,q)\}$ elements, without any restriction on the minimal cluster size.*

The theorem improves on a counterpart in the preliminary paper (Ailon et al., 2013). The consequence is, for example, that if $k_0 \le \frac{1}{2C_3(p,q)}\log n$, then the algorithm recovers with high probability clusters covering all but at most $O(n^{1/2})$ elements, without any restriction on the minimal cluster size.

3.1 Partial Observations and Active Sampling

We now consider the case where the input matrix A is not given to us in entirety, but rather that we have oracle access to $A(i,j)$ for (i,j) of our choice. Unobserved values are formally marked as $A(i,j) = *$.

Consider a more particular setting in which the edge probabilities are p' and q', and the probability of sampling an observation is ρ. More precisely: For $i \sim j$ we have $A(i,j) = 1$ with probability $\rho p'$, 0 with probability $\rho(1-p')$ and $*$ with remaining probability, independently of other pairs. For $i \not\sim j$ we have $A(i,j) = 1$ with probability $\rho q'$, 0 with probability $\rho(1-q')$ and $*$ with remaining probability, independently of other pairs. Clearly, by pretending that the values $*$ in A are 0, we emulate the full observation case of the planted partition model with parameters $p = \rho p'$, $q = \rho q'$.

Of particular interest is the case in which p', q' are held fixed and ρ tends to zero as n grows. In this regime, by varying ρ and fixing $\kappa = 1$, Theorem 2 implies the following:

Corollary 8 *There exist constants $b_3(p',q') > b_4(p',q') > 0$ and $b_5(p',q') > 0$ such that the following is true. For any sampling probability parameter $0 < \rho \leq 1$, define*

$$\ell_\sharp = b_3(p',q')\frac{\sqrt{n}}{\sqrt{\rho}} \max\left\{1, \frac{\log^4 n}{\sqrt{\rho n}}\right\}, \qquad \ell_\flat = b_4(p',q')\frac{\sqrt{n}}{\sqrt{\rho}} \ . \tag{6}$$

If for each $a \in [k]$, either $n_a \geq \ell_\sharp$ or $n_a \leq \ell_\flat$, then, with probability at least $1 - n^{-3}$, the program (CP) (after setting $*$ in A to 0) with

$$c_1 = c_1(p',q') = \frac{1}{100\sqrt{n}}\sqrt{\frac{1 - b_5(p',q')\rho}{b_5(p',q')\rho}}$$
$$c_2 = c_2(p',q') = \frac{1}{100\sqrt{n}}\sqrt{\frac{b_5(p',q')}{1 - b_5(p',q')\rho}} \ ,$$

has a unique optimal solution equal to $(\hat{K}, \hat{B}) = (\mathcal{P}_\sharp K^*, A - \hat{K})$, where $\mathcal{P}_\sharp$ is as defined in Theorem 2.

Note that we have slightly abused notation by reusing previously defined global constants (e.g., b_1) with global *functions* of p', q' (e.g., $b_1(p',q')$). Notice now that the sampling probability ρ can be used as a tuning parameter for controlling the sizes of the clusters we try to recover, instead of κ. In what follows, we will always assume the following bound on the observation rate:

$$\rho \geq \frac{\log^8 n}{n} \ , \tag{7}$$

so that the definition of $\ell_\sharp$ in (6) can be replaced by the simpler:

$$\ell_\sharp = b_3(p',q')\frac{\sqrt{n}}{\sqrt{\rho}} \ . \tag{8}$$

This assumption is made for simplicity of the exposition, and a more elaborate (though tedious) derivation can be done without it.

We now present an analogue of the converse result in Theorem 5 for the partial observation setting. Our main focus is to understand the asymptotics as $\rho \to 0$.

Theorem 9 *There exist constants $C_1(p',q'), C_2(p',q') > 0$ such that the following holds with probability at least $1 - n^{-3}$. For all observation rate parameters $\rho \leq 1$, if (K, B) is an optimal solution to (CP) with c_1, c_2 as defined in Corollary 8, and additionally K is a partial clustering corresponding to $U_1, \ldots, U_r \subseteq V$, and also*

$$\sigma_{\min}(K) \geq \max\left\{ \frac{C_1(p',q')k\log n}{\rho}, \frac{C_2(p',q')\sqrt{n\log n}}{\sqrt{\rho}} \right\} , \tag{9}$$

then $U_1, \ldots, U_r$ are actual ground truth clusters, namely, there exists an injection $\phi : [r] \mapsto [k]$ such that $U_a = V_{\phi(a)}$ for each $a \in [r]$.

The proof is similar to that of Theorem 5. The necessary changes are outlined in Section 5.3. Using the same reasoning as before, we derive the following:

Theorem 10 *Let $g = (b_3(p',q')/b_4(p',q'))^2$ (with $b_3(p',q'), b_4(p',q')$ defined in Corollary 8). There exist constants $C_3(p',q')$ and $C_4(p',q')$ such that the following holds. Assume $n \geq C_3(p',q')$ and the number of clusters k is bounded by some known number $k_0 \leq C_4(p',q')\log n$. Let $\rho_0 = \frac{b_3(p',q')^2 k_0^2 \log n}{n}$. Then there exists ρ in the set $\{\rho_0, \rho_0 g, \ldots, \rho_0 g^{k_0}\}$ for which, if A is obtained with sampling rate ρ (zeroing $*$'s), then with probability at least $1 - 2n^{-3}$, any optimal solution (K, B) to (CP) with $c_1(p',q'), c_2(p',q')$ from Corollary 8 satisfies that K is a partial clustering with the property in (9).*

Note that the upper bound on k_0 ensures that ρg^{k_0} is a probability. The theorem improves on a counterpart in the preliminary paper (Ailon et al., 2013), where k_0 was smaller by a factor of $\log\log n$ compared to here. The theorem is proven, again, using a simple pigeonhole principle, noting that one of the intervals $(\ell_\flat(\rho), \ell_\sharp(\rho))$ must be disjoint from the set of cluster sizes, and there is at least one cluster of size at least n/k_0. The value of ρ_0 is chosen so that n/k_0 is larger than the RHS of (9). This theorem motivates the iterative procedure in Algorithm 3: we start with a low sampling rate ρ, which is then increased geometrically until the program (CP) returns a partial clustering.

Theorem 10 together with Corollary 8 and Theorem 9 ensures the following. On one end of the spectrum, if k_0 is a constant (and n is large enough), then with high probability Algorithm 3 recovers at least one large cluster (of size at least n/k_0) after querying no more than

$$O\left(nk_0^2(\log n)\left(\frac{b_3(p',q')}{b_4(p',q')}\right)^{2k_0}\right) \tag{10}$$

values of $A(i,j)$. On the other end of the spectrum, if $k_0 \leq \delta\log n$ and n is large enough (exponential in $1/\delta$), then Algorithm 3 recovers at least one large cluster after querying no more than $n^{1+O(\delta)}$ values of $A(i,j)$. Iteratively recovering and removing large clusters leads to Algorithm 4 with the following guarantees.

Theorem 11 *Assume an upper bound k_0 on the number of clusters k. As long as n is larger than some function of k_0, p', q', Algorithm 4 will recover, with probability at least $1 - n^{-2}$, at least one cluster of size at least n/k_0, regardless of the size of other (small) clusters. Moreover, if k_0 is a constant, then clusters covering all but a constant number of elements will be recovered with probability at least $1 - 2n^{-2}$, and the total number of observation queries is given by (10), hence almost linear.*

Algorithm 3 RecoverBigPartialObs(V, k_0) (Assume p', q' known, fixed)

require: ground set V, oracle access to $A \in \mathbb{R}^{V \times V}$, upper bound k_0 on number of clusters

$n \leftarrow |V|$
$\rho_0 \leftarrow \frac{b_3(p',q')^2 k_0^2 \log n}{n}$
$g \leftarrow b_3(p', q')^2 / b_4(p', q')^2$
for $s \in \{0, \ldots, k_0\}$ **do**
  $\rho \leftarrow \rho_0 g^s$
  obtain matrix $A \in \{0, 1, *\}^{V \times V}$ by sampling oracle at rate ρ, then zero $*$ values in A
  // (can reuse observations from previous iterations)
  $c_1(p', q'), c_2(p', q') \leftarrow$ as in Corollary 8
  $(K, B) \leftarrow$ an optimal solution to (CP)
  if K is a partial clustering matrix satisfying (9) **then**
    return induced clusters $\{U_1, \ldots, U_r\}$
  end if
end for
return $\emptyset$

Algorithm 4 RecoverPartialObs(V, k_0) (Assume p', q' known, fixed)

require: ground set V, oracle access to $A \in \mathbb{R}^{V \times V}$, upper bound k_0 on number of clusters

$\{U_1, \ldots, U_r\} \leftarrow$ RecoverBigPartialObs(V, k_0)
$V' \leftarrow [n] \setminus (U_1 \cup \cdots \cup U_r)$
if $r = 0$ **then**
  return $\emptyset$
else
  return RecoverFullObs$(V', k_0 - r) \cup \{U_1, \ldots, U_r\}$
end if

The theorem improves on a counterpart in the preliminary paper (Ailon et al., 2013), where the recovery covers all but a super-constant (in n) number of elements. Unlike previous convex relaxation based approaches for this problem, which require all cluster sizes to be of size at least roughly $\sqrt{n}$ to succeed, there is no constraint on the cluster sizes for our algorithm.

Also note that our algorithm is an *active learning* one, because more observations fall in smaller clusters which survive deeper in the recursion of Algorithm 4. This feature can lead to a significant saving in the number of queries. When small clusters of size $\tilde{\Theta}(\sqrt{n})$ are present, previous one-shot algorithms for graph clustering with partial observations (e.g., Jalali et al., 2011; Oymak and Hassibi, 2011; Chen et al., 2014a) only guarantee recovery using $O(n^2)$ queries, which is much larger than the almost linear requirement $\tilde{O}(n)$ of our active algorithm.

4. Experiments

We test our main Algorithms 2 and 4 (with subroutines Algorithms 1 and 3) on synthetic data. In all experiment reports below, we use a variant of the Alternating Direction Method of Multipliers (ADMM) to solve the semidefinite program (CP); see Lin et al. (2011); Chen et al. (2012). The main cost of ADMM is the computation of the Singular Value Decomposition (SVD) of an $n \times n$ matrix in each round. Note that one can take advantage of the sparsity of the observations to speed up the SVD (cf. Lin et al. 2011). As is discussed in previous work, and also observed empirically by us, ADMM converges linearly, so the number of SVD needed is usually small. See the references above for further discussion of the optimization issues. The overall computation time also depends on the number of recursive calls in Algorithm 2 and 4, as well as the number of iterations used in Algorithm 1 and 3 in search for suitable values for κ and ρ (using a multiplicative update rule). These two numbers are at most $O(\max(k, \log n))$ (k is the number of clusters) under the conditions of the theorems, and in our experiments they are both quite small.

In the experiments we consider simplified versions of the algorithms: we did not make an effort to compute the constants $\ell_\sharp/\ell_\flat$ defining the algorithms, creating a difficulty in exact implementation. Instead, for Algorithm 1, we start with $\kappa = 1$ and increase κ by a multiplicative factor of 1.1 in each iteration until a partial clustering matrix is found. Similarly, in Algorithm 3, the sampling rate ρ has an initial value of 0 and is increased by an additive factor of 0.025. Still, it is obvious that our experiments support our theoretical findings. A more practical "user's guide" for this method with actual constants is subject to future work.

Whenever we say that "clusters $\{V_{i_1}, V_{i_2}, \dots\}$ were recovered", we mean that a corresponding instantiation of (CP) resulted in an optimal solution (K, B) for which K was a partial clustering matrix induced by $\{V_{i_1}, V_{i_2}, \dots\}$.

4.1 Experiment 1 (Full Observation)

Consider $n = 1100$ nodes partitioned into 4 clusters $V_1, \dots, V_4$, of sizes $800, 200, 80, 20$, respectively. The graph is generated according to the planted partition model with $p = 0.5$ and $q = 0.2$, and we assume the full observation setting. We apply the simplified version of Algorithm 2 described previously, which terminates in 4 iterations using 44 seconds. The recovered clusters at each iteration are detailed in Table 1. The table also shows the values of κ adaptively chosen by the algorithm at each iteration (which happens to equal 1 throughout). We note that the first iteration of the algorithm is similar to existing convex optimization based approaches to graph clustering (Jalali et al., 2011; Oymak and Hassibi, 2011; Chen et al., 2012); the experiment shows that these approaches by itself fail to recover all the clusters in one shot, thus necessitating the iterative procedure proposed in this paper.

4.2 Experiment 2 (Partial Observation, Fixed Sample Rate)

We have $n = 1100$ with clusters $V_1, \dots, V_4$ of sizes $800, 200, 50, 50$. The observed graph is generated with $p' = 0.7$, $q' = 0.1$, and observation rate $\rho = 0.3$. We repeatedly solve (CP) with c_1, c_2 given in Corollary 8. At each iteration, we see that at least one large

Iteration	κ	# nodes left	Clusters recovered
1	1	1100	V_1
2	1	300	V_2
3	1	100	V_3
4	1	20	V_4

Table 1: Results for experiment 1: $n = 1100$, $\{|V_a|\} = \{800, 200, 80, 20\}$, $p = 0.5$, $q = 0.2$, fixed $\rho = 1$.

Iteration	κ	# nodes left	Clusters recovered
1	1	1100	V_1
2	1	300	V_2
3	1	100	V_3, V_4

Table 2: Results for experiment 2: $n = 1100$, $\{|V_a|\} = \{800, 200, 50, 50\}$, $p' = 0.7$, $q' = 0.1$, fixed $\rho = 0.3$.

cluster (compared to the input size at that iteration) is recovered exactly and removed. The experiment terminates in 3 iterations using 18 seconds. Results are shown in Table 2.

4.3 Experiment 3 (Partial Observation, Adaptive Sampling Rate)

We use the simplified version of Algorithm 4 described previously. We have $n = 1100$ with clusters $V_1, \ldots, V_4$ of sizes $800, 200, 50, 50$. The graph is generated with $p' = 0.8$ and $q' = 0.2$, and then adaptively sampled by the algorithm. The algorithm terminates in 3 iterations using 148 seconds. Table 3 shows the recovery result and the sampling rates used in each iteration. From the table we can see that the expected total number of observed entries used by the algorithm is

$$1100^2 \cdot 0.125 + 300^2 \cdot 0.25 + 100^2 \cdot 0.55 = 179250,$$

which is 14.8% of all possible node pairs (the actual number of observations is very close to this expected value). In comparison, we perform another experiment using a non-adaptive sampling rate, for which we need $\rho = 97.5\%$ in order to recover all the clusters in one shot. Therefore, our adaptive algorithm achieves a significant saving in the number of queries.

4.4 Experiment 3A

We repeat the above experiment with a larger instance: $n = 4500$ with clusters $V_1, \ldots, V_6$ of sizes $3200, 800, 200, 200, 50, 50$, and $p' = 0.8, q' = 0.2$. The algorithm terminates in 182 seconds, with results shown in Table 4. Note that we recover the smallest clusters, whose sizes are below $\sqrt{n}$. The expected total number of observations used by the algorithm is 3388000, which is 16.7% of all possible node pairs. Using a non-adaptive sampling rate

Iteration	ρ	# nodes left	Clusters recovered
1	0.125	1100	V_1
2	0.25	300	V_2
3	0.55	100	V_3, V_4

Table 3: Results for experiment 3: $n = 1100$, $\{|V_a|\} = \{800, 200, 50, 50\}$, $p' = 0.8$, $q' = 0.2$.

Iteration	ρ	# nodes left	Clusters recovered
1	0.15	4500	V_1
2	0.175	1300	V_2
3	0.2	500	V_3, V_4
4	0.475	100	V_5, V_6

Table 4: Results for experiment 3A: $n = 4500$, $\{|V_a|\} = \{3200, 800, 200, 200, 50, 50\}$, $p' = 0.8$, $q' = 0.2$.

$\rho = 35.0\%$ only recovers the 4 largest clusters, and we are unable to recover all 6 clusters in one shot even with $\rho = 1$.

4.5 Experiment 4 (Mid-Size Clusters)

Our current theoretical results do not say anything about the mid-size clusters—those with sizes between $\ell_\flat$ and $\ell_\sharp$. It is interesting to investigate the behavior of (CP) in the presence of mid-size clusters. We generate an instance with $n = 750$ nodes partitioned into four clusters of sizes $\{500, 150, 70, 30\}$, edge probabilities $p = 0.8, q = 0.2$ and a sampling rate $\rho = 0.12$. We then solve (CP) with a fixed $\kappa = 1$. The low-rank part K of the solution is shown in Figure 2. The large cluster of size 500 is completely recovered in K, while the two small clusters of sizes 70 and 30 are entirely ignored. The medium cluster of size 150, however, exhibits a pattern we find difficult to characterize. This shows that the constant gap between $\ell_\sharp$ and $\ell_\flat$ in our theorems is a real phenomenon and not an artifact of our proof techniques. Nevertheless, the mid-size cluster appears clean, and might allow recovery using a simple combinatorial procedure. If this is true in general, it might not be necessary to search for a gap free of cluster sizes. In particular, perhaps for any κ, (CP) identifies all large clusters above $\ell_\sharp$ after a simple mid-size cleanup procedure, and ignores all other clusters. Understanding this phenomenon and its algorithmic implications is of much interest.

5. Proofs

We use the following notation and conventions throughout the proofs. *With high probability* or *w.h.p.* means with probability at least $1 - n^{-6}$. The expressions $a \vee b$ and $a \wedge b$ mean $\max\{a, b\}$ and $\min\{a, b\}$, respectively. For a real $n \times n$ matrix M, we use the unadorned norm $\|M\|$ to denote its spectral norm. The notation $\|M\|_F$ refers to the Frobenius norm,

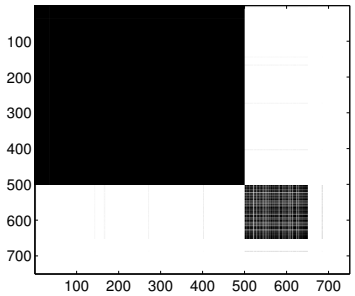


Figure 2: The solution to (CP) with mid-size clusters.

$\|M\|_1$ is $\sum_{i,j} |M(i,j)|$, and $\|M\|_\infty$ is $\max_{ij} |M(i,j)|$. We shall use the standard inner product $\langle X, Y\rangle := \sum_{i,j=1}^{n} X(i,j)Y(i,j)$.

We will also study operators on the space of matrices, and denote them using a calligraphic font, e.g., $\mathcal{P}$. The norm $\|\mathcal{P}\|$ of an operator is defined as

$$\|\mathcal{P}\| := \sup_{M \in \mathbb{R}^{n\times n}: \|M\|_F = 1} \|\mathcal{P}M\|_F \ .$$

For a fixed real $n \times n$ matrix M, we define the matrix linear subspace $T(M)$ as follows:

$$T(M) := \{YM + MX : X, Y \in \mathbb{R}^{n\times n}\} \ .$$

In words, this subspace is the set of matrices spanned by matrices each row of which is in the row space of M, and matrices each column of which is in the column space of M. We let $T(M)^\perp$ denote the orthogonal subspace to $T(M)$ with respect to $\langle \cdot, \cdot \rangle$, which is given by

$$T(M)^\perp := \{X \in \mathbb{R}^{n\times n} : \langle X, Y\rangle = 0, \forall Y \in T(M)\} \ .$$

It is a well known fact that the projection $\mathcal{P}_{T(X)}$ onto $T(X)$ w.r.t. $\langle \cdot, \cdot \rangle$ is given by

$$\mathcal{P}_{T(X)}M := \mathcal{P}_{C(X)}M + \mathcal{P}_{R(X)}M - \mathcal{P}_{C(X)}\mathcal{P}_{R(X)}M \ ,$$

where $\mathcal{P}_{C(X)}$ is projection (of each column of a matrix) onto the column space of X, and $\mathcal{P}_{R(X)}$ is projection onto the row space of X. The projection onto $T(M)^\perp$ is $\mathcal{P}_{T(X)^\perp}M = M - \mathcal{P}_{T(X)}M$.

Finally, we recall that $\mathrm{s}(M)$ is the support of M, $\mathcal{P}_{\mathrm{s}(M)}X$ is the matrix obtained from X by setting its entries outside $\mathrm{s}(M)$ to zero, and $\mathcal{P}_{\mathrm{s}(M)^c}X := X - \mathcal{P}_{\mathrm{s}(M)}X$.

5.1 Proof of Theorem 2

The proof builds on the analysis in Chen et al. (2012). We need some additional notation:

1. We let $V_\flat \subseteq V$ denote the set of of elements i such that $n_{\langle i\rangle} \le \ell_\flat$. (We remind the reader that $n_{\langle i\rangle} = |V_{\langle i\rangle}|$.)

2. We remind the reader that the projection $\mathcal{P}_\sharp$ is defined as follows:

$$(\mathcal{P}_\sharp M)(i,j) = \begin{cases} M(i,j), & \max\{n_{\langle i\rangle}, n_{\langle j\rangle}\} \ge \ell_\sharp \\ 0, & \text{otherwise.} \end{cases}$$

3. The projection $\mathcal{P}_\flat$ is defined as follows:

$$(\mathcal{P}_\flat M)(i,j) = \begin{cases} M(i,j), & \max\{n_{\langle i\rangle}, n_{\langle j\rangle}\} \le \ell_\flat \\ 0, & \text{otherwise.} \end{cases}$$

In words, $\mathcal{P}_\flat$ projects onto the set of matrices supported on $V_\flat \times V_\flat$. Note that by the theorem assumption, $\mathcal{P}_\sharp + \mathcal{P}_\flat = \mathcal{I}d$ (equivalently, $\mathcal{P}_\sharp$ projects onto the set of matrices supported on $(V \times V) \setminus (V_\flat \times V_\flat)$).

4. We use $U\Sigma U^\top$ to denote the rank-k' Singular Value Decomposition (SVD) of the symmetric matrix $\hat{K}$, where $k' = \mathrm{rank}(\hat{K})$ and equals the number of clusters with size at least $\ell_\sharp$.

5. Define the set

$$\mathfrak{D} := \left\{\Delta \in \mathbb{R}^{n\times n} | \Delta_{ij} \le 0, \forall i \sim j, (i,j) \notin V_\flat \times V_\flat; 0 \le \Delta_{ij}, \forall i \not\sim j, (i,j) \notin V_\flat \times V_\flat\right\},$$

which strictly contains all feasible deviation from $\hat{K}$.

6. For simplicity we write $T := T(\hat{K})$.

We will make use of the following facts:

1. $\mathcal{I}d = \mathcal{P}_{\mathrm{s}(\hat{B})} + \mathcal{P}_{\mathrm{s}(\hat{B})^c} = \mathcal{P}_{\mathrm{s}(A)} + \mathcal{P}_{\mathrm{s}(A)^c}$.
2. $\mathcal{P}_\sharp, \mathcal{P}_\flat, \mathcal{P}_{\mathrm{s}(\hat{B})}, \mathcal{P}_{\mathrm{s}(\hat{B})^c}, \mathcal{P}_{\mathrm{s}(A)}$, and $\mathcal{P}_{\mathrm{s}(A)^c}$ commute with each other.

5.1.1 Approximate Dual Certificate Condition

We begin by giving a deterministic sufficient condition for $(\hat{K}, \hat{B})$ to be the unique optimal solution to the program (CP).

Proposition 12 *$(\hat{K}, \hat{B})$ is the unique optimal solution to (CP) if there exists a matrix $Q \in \mathbb{R}^{n\times n}$ and a number $0 < \epsilon < 1$ satisfying:*

1. $\|Q\| < 1$;
2. $\|\mathcal{P}_T(Q)\|_\infty \le \frac{\epsilon}{2}\min\{c_1, c_2\}$;
3. $\forall \Delta \in \mathfrak{D}$:
   (a) $\left\langle UU^\top + Q, \mathcal{P}_{\mathrm{s}(A)}\mathcal{P}_{\mathrm{s}(\hat{B})}\mathcal{P}_\sharp\Delta\right\rangle = (1+\epsilon)c_1\left\|\mathcal{P}_{\mathrm{s}(A)}P_{\mathrm{s}(\hat{B})}\mathcal{P}_\sharp\Delta\right\|_1$,
   (b) $\left\langle UU^\top + Q, \mathcal{P}_{\mathrm{s}(A)^c}\mathcal{P}_{\mathrm{s}(\hat{B})}\mathcal{P}_\sharp\Delta\right\rangle = (1+\epsilon)c_2\left\|\mathcal{P}_{\mathrm{s}(A)^c}P_{\mathrm{s}(\hat{B})}\mathcal{P}_\sharp\Delta\right\|_1$;
4. $\forall \Delta \in \mathfrak{D}$:
   (a) $\left\langle UU^\top + Q, \mathcal{P}_{\mathrm{s}(A)}\mathcal{P}_{\mathrm{s}(\hat{B})^c}\mathcal{P}_\sharp\Delta\right\rangle \ge -(1-\epsilon)c_1\left\|\mathcal{P}_{\mathrm{s}(A)}P_{\mathrm{s}(\hat{B})^c}\mathcal{P}_\sharp\Delta\right\|_1$,
   (b) $\left\langle UU^\top + Q, \mathcal{P}_{\mathrm{s}(A)^c}\mathcal{P}_{\mathrm{s}(\hat{B})^c}\mathcal{P}_\sharp\Delta\right\rangle \ge -(1-\epsilon)c_2\left\|\mathcal{P}_{\mathrm{s}(A)^c}P_{\mathrm{s}(\hat{B})^c}\mathcal{P}_\sharp\Delta\right\|_1$;

5. $\mathcal{P}_{\mathrm{s}(\hat{B})}\mathcal{P}_\flat(UU^\top + Q) = c_1\mathcal{P}_\flat\hat{B}$;

6. $\left\|\mathcal{P}_{\mathrm{s}(\hat{B})^c}\mathcal{P}_\flat(UU^\top + Q)\right\|_\infty \le c_2$.

Proof Consider any feasible solution $(\hat{K}+\Delta, \hat{B}-\Delta)$ to (CP); we know $\Delta \in \mathfrak{D}$ due to the inequality constraints in (CP). We will show that this solution will have strictly higher objective value than $(\hat{K}, \hat{B})$ if $\Delta \neq 0$.

For this Δ, let G_Δ be a matrix in $T^\perp \cap \mathrm{Range}(\mathcal{P}_\flat)$ satisfying $\|G_\Delta\| = 1$ and $\langle G_\Delta, \Delta\rangle = \|\mathcal{P}_{T^\perp}\mathcal{P}_\flat\Delta\|_*$; such a matrix always exists because $\mathrm{Range}\mathcal{P}_\flat \subseteq T^\perp$. Suppose $\|Q\| = b$. Clearly, $\mathcal{P}_{T^\perp}Q + (1-b)G_\Delta \in T^\perp$ and, due to Property 1 in the proposition, we have $b < 1$ and $\|\mathcal{P}_{T^\perp}Q + (1-b)G_\Delta\| \le \|Q\| + (1-b)\|G_\Delta\| = b + (1-b) = 1$. Therefore, $UU^\top + \mathcal{P}_{T^\perp}Q + (1-b)G_\Delta$ is a subgradient of $f(K) = \|K\|_*$ at $K = \hat{K}$. On the other hand, define the matrix $F_\Delta = -\mathcal{P}_{\mathrm{s}(\hat{B})^c}\mathrm{sgn}(\Delta)$. We have $F_\Delta \in \mathrm{s}(\hat{B})^c$ and $\|F_\Delta\|_\infty \le 1$. Therefore, $\mathcal{P}_{\mathrm{s}(A)}(\hat{B} + F_\Delta)$ is a subgradient of $g_1(B) = \left\|\mathcal{P}_{\mathrm{s}(A)}B\right\|_1$ at $B = \hat{B}$, and $\mathcal{P}_{\mathrm{s}(A)^c}(\hat{B} + F_\Delta)$ is a subgradient of $g_2(B) = \left\|\mathcal{P}_{\mathrm{s}(A)^c}B\right\|_1$ at $B = \hat{B}$. Using these three subgradients, the difference in the objective value can be bounded as follows:

$$
\begin{aligned}
&d(\Delta)\\
\triangleq&\left\|\hat{K}+\Delta\right\|_* + c_1\left\|\mathcal{P}_{\mathrm{s}(A)}(\hat{B}-\Delta)\right\|_1 + c_2\left\|\mathcal{P}_{\mathrm{s}(A)^c}(\hat{B}-\Delta)\right\|_1 - \left\|\hat{K}\right\|_* - c_1\left\|\mathcal{P}_{\mathrm{s}(A)}\hat{B}\right\|_1\\
&- c_2\left\|\mathcal{P}_{\mathrm{s}(A)^c}\hat{B}\right\|_1\\
\ge&\left\langle UU^\top + \mathcal{P}_{T^\perp}Q + (1-b)G_\Delta, \Delta\right\rangle + c_1\left\langle\mathcal{P}_{\mathrm{s}(A)}(\hat{B}+F_\Delta), -\Delta\right\rangle + c_2\left\langle\mathcal{P}_{\mathrm{s}(A)^c}(\hat{B}+F_\Delta), -\Delta\right\rangle\\
=&(1-b)\|\mathcal{P}_{T^\perp}\mathcal{P}_\flat\Delta\|_* + \left\langle UU^\top + \mathcal{P}_{T^\perp}Q, \Delta\right\rangle + c_1\left\langle\mathcal{P}_{\mathrm{s}(A)}\hat{B}, -\Delta\right\rangle + c_2\left\langle\mathcal{P}_{\mathrm{s}(A)^c}\hat{B}, -\Delta\right\rangle\\
&+ c_1\left\langle\mathcal{P}_{\mathrm{s}(A)}F_\Delta, -\Delta\right\rangle + c_2\left\langle\mathcal{P}_{\mathrm{s}(A)^c}F_\Delta, -\Delta\right\rangle\\
=&(1-b)\|\mathcal{P}_{T^\perp}\mathcal{P}_\flat\Delta\|_* + \left\langle UU^\top + \mathcal{P}_{T^\perp}Q, \Delta\right\rangle + c_1\left\langle\mathcal{P}_\flat\mathcal{P}_{\mathrm{s}(A)}\hat{B}, -\Delta\right\rangle + c_2\left\langle\mathcal{P}_\flat\mathcal{P}_{\mathrm{s}(A)^c}\hat{B}, -\Delta\right\rangle\\
&+ c_1\left\langle\mathcal{P}_\sharp\mathcal{P}_{\mathrm{s}(A)}\hat{B}, -\Delta\right\rangle + c_2\left\langle\mathcal{P}_\sharp\mathcal{P}_{\mathrm{s}(A)^c}\hat{B}, -\Delta\right\rangle + c_1\left\langle\mathcal{P}_{\mathrm{s}(A)}F_\Delta, -\Delta\right\rangle + c_2\left\langle\mathcal{P}_{\mathrm{s}(A)^c}F_\Delta, -\Delta\right\rangle.
\end{aligned}
$$

The last six terms of the last RHS satisfy:

1. $c_1\left\langle\mathcal{P}_\flat\mathcal{P}_{\mathrm{s}(A)}\hat{B}, -\Delta\right\rangle + c_2\left\langle\mathcal{P}_\flat\mathcal{P}_{\mathrm{s}(A)^c}\hat{B}, -\Delta\right\rangle = c_1\left\langle\mathcal{P}_\flat\hat{B}, -\Delta\right\rangle$, because $\mathcal{P}_\flat\hat{B} \in \mathrm{s}(A)$.

2. $\left\langle\mathcal{P}_\sharp\mathcal{P}_{\mathrm{s}(A)}\hat{B}, -\Delta\right\rangle \ge -\left\|\mathcal{P}_\sharp\mathcal{P}_{\mathrm{s}(A)}\mathcal{P}_{\mathrm{s}(\hat{B})}\Delta\right\|_1$ and $\left\langle\mathcal{P}_\sharp\mathcal{P}_{\mathrm{s}(A)^c}\hat{B}, \Delta\right\rangle \ge -\left\|\mathcal{P}_\sharp\mathcal{P}_{\mathrm{s}(A)^c}\mathcal{P}_{\mathrm{s}(\hat{B})}\Delta\right\|_1$, because $\hat{B} \in \mathrm{s}(\hat{B})$ and $\left\|\hat{B}\right\|_\infty \le 1$.

3. $\left\langle\mathcal{P}_{\mathrm{s}(A)}F_\Delta, -\Delta\right\rangle = \left\|\mathcal{P}_{\mathrm{s}(A)}\mathcal{P}_{\mathrm{s}(\hat{B})^c}\Delta\right\|_1$ and $\left\langle\mathcal{P}_{\mathrm{s}(A)^c}F_\Delta, -\Delta\right\rangle = \left\|\mathcal{P}_{\mathrm{s}(A)^c}\mathcal{P}_{\mathrm{s}(\hat{B})^c}\Delta\right\|_1$, due to the definition of F.

It follows that

$$
\begin{aligned}
d(\Delta) \ge&(1-b)\|\mathcal{P}_{T^\perp}\mathcal{P}_\flat\Delta\|_* + \left\langle UU^\top + \mathcal{P}_{T^\perp}Q, \Delta\right\rangle + c_1\left\langle\mathcal{P}_\flat\hat{B}, -\Delta\right\rangle - c_1\left\|\mathcal{P}_\sharp\mathcal{P}_{\mathrm{s}(A)}\mathcal{P}_{\mathrm{s}(\hat{B})}\Delta\right\|_1\\
&- c_2\left\|\mathcal{P}_\sharp\mathcal{P}_{\mathrm{s}(A)^c}\mathcal{P}_{\mathrm{s}(\hat{B})}\Delta\right\|_1 + c_1\left\|\mathcal{P}_{\mathrm{s}(A)}\mathcal{P}_{\mathrm{s}^c(\hat{B})}\Delta\right\|_1 + c_2\left\|\mathcal{P}_{\mathrm{s}(A)^c}\mathcal{P}_{\mathrm{s}^c}\Delta\right\|_1. \qquad (11)
\end{aligned}
$$

Consider the second term in the last RHS, which equals

$$\left\langle UU^\top + \mathcal{P}_{T^\perp} Q, \Delta \right\rangle = \left\langle UU^\top + Q, \mathcal{P}_\sharp \Delta \right\rangle + \left\langle UU^\top + Q, \mathcal{P}_\flat \Delta \right\rangle - \langle \mathcal{P}_T Q, \Delta \rangle .$$

We bound these three terms separately. For the first term, we have

$$\begin{aligned}
&\left\langle UU^\top + Q, \mathcal{P}_\sharp \Delta \right\rangle \\
=&\left\langle UU^\top + Q, \left(\mathcal{P}_{\mathrm{s}(A)} \mathcal{P}_{\mathrm{s}(\hat{B})} \mathcal{P}_\sharp + \mathcal{P}_{\mathrm{s}(A)^c} \mathcal{P}_{\mathrm{s}(\hat{B})} \mathcal{P}_\sharp + \mathcal{P}_{\mathrm{s}(A)} \mathcal{P}_{\mathrm{s}(\hat{B})^c} \mathcal{P}_\sharp + \mathcal{P}_{\mathrm{s}(A)^c} \mathcal{P}_{\mathrm{s}^c} \mathcal{P}_\sharp \right) \Delta \right\rangle \\
\geq&(1+\epsilon) c_1 \left\| \mathcal{P}_{\mathrm{s}(A)} P_{\mathrm{s}(\hat{B})} \mathcal{P}_\sharp \Delta \right\|_1 + (1+\epsilon) c_2 \left\| \mathcal{P}_{\mathrm{s}(A)^c} P_{\mathrm{s}(\hat{B})} \mathcal{P}_\sharp \Delta \right\|_1 - (1-\epsilon) c_1 \left\| \mathcal{P}_{\mathrm{s}(A)} P_{\mathrm{s}(\hat{B})^c} \mathcal{P}_\sharp \Delta \right\|_1 \\
&- (1-\epsilon) c_2 \left\| \mathcal{P}_{\mathrm{s}(A)^c} P_{\mathrm{s}(\hat{B})^c} \mathcal{P}_\sharp \Delta \right\|_1 . \quad \text{(Using Properties 3 and 4.)}
\end{aligned}$$

For the second term, we have

$$\begin{aligned}
&\left\langle UU^\top + Q, \mathcal{P}_\flat \Delta \right\rangle \\
=&\left\langle \mathcal{P}_{\mathrm{s}(\hat{B})} \mathcal{P}_\flat (UU^\top + Q), \Delta \right\rangle + \left\langle \mathcal{P}_{\mathrm{s}(\hat{B})^c} \mathcal{P}_\flat (UU^\top + Q), \Delta \right\rangle \\
\geq&c_1 \left\langle \mathcal{P}_\flat \hat{B}, \Delta \right\rangle - c_2 \left\| \mathcal{P}_{\mathrm{s}(\hat{B})^c} \mathcal{P}_\flat \Delta \right\|_1 \quad \text{(using Properties 5 and 6)} \\
=&c_1 \left\langle \mathcal{P}_\flat \hat{B}, \Delta \right\rangle - c_2 \left\| \mathcal{P}_{\mathrm{s}(A)^c} \mathcal{P}_{\mathrm{s}(\hat{B})^c} \mathcal{P}_\flat \Delta \right\|_1 . \quad \text{(Because } \mathcal{P}_{\mathrm{s}(A)^c} \mathcal{P}_{\mathrm{s}(\hat{B})^c} \mathcal{P}_\flat = \mathcal{P}_{\mathrm{s}(\hat{B})^c} \mathcal{P}_\flat .)
\end{aligned}$$

Finally, for the third term, Due to the block diagonal structure of the elements of T, we have $\mathcal{P}_T = \mathcal{P}_\sharp \mathcal{P}_T$ and therefore

$$\langle -\mathcal{P}_T Q, \Delta \rangle = - \langle \mathcal{P}_T Q, \mathcal{P}_\sharp \Delta \rangle \geq - \left\| \mathcal{P}_T Q \right\|_\infty \left\| \mathcal{P}_\sharp \Delta \right\|_1 \geq - \frac{\epsilon}{2} \min \{c_1, c_2\} \left\| \mathcal{P}_\sharp \Delta \right\|_1 .$$

Combining the above three bounds with Eq. (11), we obtain

$$\begin{aligned}
&d(\Delta) \\
\geq&(1-b) \left\| \mathcal{P}_{T^\perp} \mathcal{P}_\flat \Delta \right\|_* + \epsilon c_1 \left\| \mathcal{P}_\sharp \mathcal{P}_{\mathrm{s}(A)} \mathcal{P}_{\mathrm{s}(\hat{B})} \Delta \right\|_1 + \epsilon c_2 \left\| \mathcal{P}_\sharp \mathcal{P}_{\mathrm{s}(A)^c} \mathcal{P}_{\mathrm{s}(\hat{B})} \Delta \right\|_1 + \epsilon c_1 \left\| \mathcal{P}_{\mathrm{s}(A)} \mathcal{P}_{\mathrm{s}(\hat{B})^c} \mathcal{P}_\sharp \Delta \right\|_1 \\
&+ \epsilon c_2 \left\| \mathcal{P}_{\mathrm{s}(A)^c} \mathcal{P}_{\mathrm{s}(\hat{B})^c} \mathcal{P}_\sharp \Delta \right\|_1 + c_1 \left\| \mathcal{P}_{\mathrm{s}(A)} \mathcal{P}_{\mathrm{s}(\hat{B})^c} \mathcal{P}_\flat \Delta \right\|_1 - \frac{\epsilon}{2} \min \{c_1, c_2\} \left\| \mathcal{P}_\sharp \Delta \right\|_1 \\
=&(1-b) \left\| \mathcal{P}_{T^\perp} \mathcal{P}_\flat \Delta \right\|_* + \epsilon c_1 \left\| \mathcal{P}_\sharp \mathcal{P}_{\mathrm{s}(A)} \Delta \right\|_1 + \epsilon c_2 \left\| \mathcal{P}_\sharp \mathcal{P}_{\mathrm{s}(A)^c} \Delta \right\|_1 - \frac{\epsilon}{2} \min \{c_1, c_2\} \left\| \mathcal{P}_\sharp \Delta \right\|_1 \\
&\text{(note that } \mathcal{P}_{\mathrm{s}(A)} \mathcal{P}_{\mathrm{s}(\hat{B})^c} \mathcal{P}_\flat \Delta = 0) \\
\geq&(1-b) \left\| \mathcal{P}_\flat \Delta \right\|_* + \frac{\epsilon}{2} \min \{c_1, c_2\} \left\| \mathcal{P}_\sharp \Delta \right\|_1 ,
\end{aligned}$$

which is strictly greater than zero for $\Delta \neq 0$. ■

5.1.2 Constructing Q

To prove the theorem, it suffices to show that with probability at least $1 - n^{-3}$, there exists a matrix Q with the properties required by Proposition 12. We do this by explicitly

constructing Q. Suppose we take

$$\epsilon := \frac{100}{\sqrt{t(1-t)}} \max\left\{ \frac{\kappa\sqrt{n}}{\ell_\sharp}, \sqrt{\frac{\log^4 n}{\ell_\sharp}} \right\},$$

and use the weights c_1 and c_2 given in Theorem 2. We specify $\mathcal{P}_\sharp Q$ and $\mathcal{P}_\flat Q$ separately.

The matrix $\mathcal{P}_\sharp Q$ is given by $\mathcal{P}_\sharp Q = \mathcal{P}_\sharp Q_1 + \mathcal{P}_\sharp Q_2 + \mathcal{P}_\sharp Q_3$, where for $(i,j) \notin V_\flat \times V_\flat$,

$$\mathcal{P}_\sharp Q_1(i,j) = \begin{cases} -\frac{1}{n_{\langle i \rangle}}, & i \sim j, (i,j) \in \mathrm{s}(\hat{B}) \\ \frac{1}{n_{\langle i \rangle}} \cdot \frac{1-p_{ij}}{p_{ij}}, & i \sim j, (i,j) \in \mathrm{s}(\hat{B})^c \\ 0, & i \not\sim j \end{cases}$$

$$\mathcal{P}_\sharp Q_2(i,j) = \begin{cases} -(1+\epsilon)c_2, & i \sim j, (i,j) \in \mathrm{s}(\hat{B}) \\ (1+\epsilon)c_2 \frac{1-p_{ij}}{p_{ij}}, & i \sim j, (i,j) \in \mathrm{s}(\hat{B})^c \\ 0, & i \not\sim j \end{cases}$$

$$\mathcal{P}_\sharp Q_3(i,j) = \begin{cases} (1+\epsilon)c_1, & i \not\sim j, (i,j) \in \mathrm{s}(\hat{B}) \\ -(1+\epsilon)c_1 \frac{q_{ij}}{1-q_{ij}}, & i \not\sim j, (i,j) \in \mathrm{s}(\hat{B})^c \\ 0, & i \sim j. \end{cases}$$

Note that these matrices have zero-mean entries. (Recall that $\mathrm{s}(\hat{B}) = \mathrm{s}(A - \hat{K})$ is a random set since the graph A is random.)

$\mathcal{P}_\flat Q$ is given as follows. For $(i,j) \in V_\flat \times V_\flat$,

$$\mathcal{P}_\flat Q(i,j) = \begin{cases} c_1, & i \sim j, (i,j) \in \mathrm{s}(A) \\ -c_2, & i \sim j, (i,j) \in \mathrm{s}(A)^c \\ c_1, & i \not\sim j, (i,j) \in \mathrm{s}(A) \\ c_2 W(i,j), & i \not\sim j, (i,j) \in \mathrm{s}(A)^c, \end{cases}$$

where W is a symmetric matrix whose upper-triangle entries are independent and obey

$$W(i,j) = \begin{cases} +1, & \text{with probability } \frac{t-q}{2t(1-q)}, \\ -1, & \text{with remaining probability.} \end{cases}$$

Note that we introduced additional randomness in W.

5.1.3 Validating Q

Under the choice of t in Theorem 2, we have $\frac{1}{4}p \le t \le p$ and $\frac{1}{4}(1-q) \le 1-t \le 1-q$. Also under the assumption (1) in the theorem and since $p - q \le p(1-q)$, $\ell_\sharp \le n$, we have $p(1-q) \ge \frac{b_3^2 \kappa^2 n}{\ell_\sharp^2} \vee \frac{b_3 \log^4 n}{\ell_\sharp} \ge \frac{b_3 \log^4 n}{n}$. Using these inequalities, it is easy to check that $\epsilon < \frac{1}{2}$ provided that the constant b_3 is sufficiently large. We will make use of these facts frequently in the proof.

We now verify that the Q constructed above satisfy the six properties in Proposition 12 with probability at least $1 - n^{-3}$.

Property 1:

Suppose the matrix $Q_\sim$ is obtained from Q by setting all $Q(i,j)$ with $i \not\sim j$ to zero, and $Q_{\not\sim} = Q - Q_\sim$. Note that $\|Q\| \le \|\mathcal{P}_\sharp Q_\sim\| + \|\mathcal{P}_\sharp Q_{\not\sim}\| + \|\mathcal{P}_\flat Q_\sim\| + \|\mathcal{P}_\flat Q_{\not\sim}\|$. Below we show that with high probability, the first term is upper-bounded by $\frac{7}{32}$ and the other threes terms are upper-bounded by $\frac{1}{4}$, which establishes that $\|Q\| \le \frac{31}{32}$.

(a) $\mathcal{P}_\flat Q_\sim$ is a block diagonal matrix support on $V_\flat \times V_\flat$, where the size of each block is at most $\ell_\flat$. Note that $\mathcal{P}_\flat Q_\sim = \mathbb{E}[\mathcal{P}_\flat Q_\sim] + (\mathcal{P}_\flat Q_\sim - \mathbb{E}[\mathcal{P}_\flat Q_\sim])$. Here $\mathbb{E}[\mathcal{P}_\flat Q_\sim]$ is a deterministic matrix with all non-zero entries equal to $\frac{1}{100\kappa\sqrt{n}}\frac{p-t}{\sqrt{t(1-t)}}$. We thus have

$$\|\mathbb{E}[\mathcal{P}_\flat Q_\sim]\| \le \ell_\flat \frac{1}{100\kappa\sqrt{n}}\frac{p-t}{\sqrt{t(1-t)}} \le \frac{1}{32},$$

where the last inequality holds under the definition of $\ell_\flat$ in Theorem 2. On the other hand, the matrix $\mathcal{P}_\flat Q_\sim - \mathbb{E}[\mathcal{P}_\flat Q_\sim]$ is a random matrix whose entries are independent, bounded almost surely by $B := \max\{c_1, c_2\}$ and have zero mean with variance bounded by $\frac{1}{100^2\kappa^2 n} \cdot \frac{p(1-p)}{t(1-t)}$. If $\ell_\flat \le n^{2/3}$, we apply part 1 of Lemma 17 to obtain

$$\begin{aligned}\|\mathcal{P}_\flat Q_\sim - \mathbb{E}[\mathcal{P}_\flat Q_\sim]\| &\le 10\max\left\{\frac{1}{100\kappa\sqrt{n}}\sqrt{\frac{p(1-p)}{t(1-t)}\ell_\flat \log n}, (c_1 \vee c_2)\log n\right\} \\ &\le \max\left\{\frac{1}{10\kappa}\sqrt{\frac{p(1-p)}{t(1-t)}\frac{\log n}{n^{1/3}}}, \frac{1}{10\kappa\sqrt{n}}\left(\sqrt{\frac{1-t}{t}} \vee \sqrt{\frac{t}{1-t}}\right)\log n\right\} \le \frac{3}{16},\end{aligned}$$

where the last inequality follows from $t(1-t) \ge \frac{p(1-q)}{16} \gtrsim \frac{\log^4 n}{n}$. If $\ell_\flat \ge n^{2/3} \ge 76$, then the variance of the entries is bounded by $\sigma^2 := \frac{1}{100^2\kappa^2 nt(1-t)}\left(p(1-p) \vee \frac{t^2\log^4 n}{\ell_\flat} \vee \frac{(1-t)^2\log^4 n}{\ell_\flat}\right)$, and $\sigma \gtrsim \frac{B\log^2 n}{\sqrt{\ell_\flat}}$. Hence we can apply part 2 of Lemma 17 to get

$$\|\mathcal{P}_\flat Q_\sim - \mathbb{E}[\mathcal{P}_\flat Q_\sim]\| \le 10\sigma\sqrt{\ell_\flat} \le \frac{3}{16}, \text{w.h.p.},$$

where in the last inequality we use $n \ge \ell_\flat$ and $t(1-t) \ge \frac{1}{16}p(1-q) \gtrsim \frac{\log^4 n}{n}$. We conclude that $\|\mathcal{P}_\flat Q_\sim\| \le \|\mathbb{E}[\mathcal{P}_\flat Q_\sim]\| + \|\mathcal{P}_\flat Q_\sim - \mathbb{E}[\mathcal{P}_\flat Q_\sim]\| \le \frac{1}{32} + \frac{3}{16} = \frac{7}{32}$ w.h.p.

(b) $\mathcal{P}_\flat Q_{\not\sim}$ is a random matrix supported on $V_\flat \times V_\flat$, whose entries are independent, zero mean, bounded almost surely by $B' := \max\{c_1, c_2\}$, and have variance $\frac{1}{100^2\kappa^2 n} \cdot \frac{t^2+q-2tq}{(1-t)t}$. If $\ell_\flat \le n^{2/3}$, we apply part 1 of Lemma 17 to obtain

$$\begin{aligned}\|\mathcal{P}_\flat Q_{\not\sim}\| &\le 10\max\left\{\frac{1}{100\kappa\sqrt{n}}\sqrt{\frac{t^2+q-2tq}{t(1-t)}\ell_\flat \log n}, (c_1 \vee c_2)\log n\right\} \\ &\le \max\left\{\frac{1}{10\kappa}\sqrt{\frac{t^2+q-2tq}{t(1-t)}\frac{\log n}{n^{1/3}}}, \frac{1}{10\kappa\sqrt{n}}\left(\sqrt{\frac{1-t}{t}} \vee \sqrt{\frac{t}{1-t}}\right)\log n\right\} \le \frac{1}{4},\end{aligned}$$

where the last inequality follows from $t(1-t) \ge \frac{p(1-q)}{16} \gtrsim \frac{\log^4 n}{n}$. If $\ell_\flat \ge n^{2/3} \ge 76$, one verifies that the variance of the entries is bounded by $(\sigma')^2 := \frac{1}{100^2\kappa^2 n}\cdot\left(\frac{t^2+q-2tq}{(1-t)t} \vee \frac{t\log^4 n}{(1-t)\ell_\flat} \vee \frac{(1-t)\log^4 n}{t\ell_\flat}\right)$,

and $\sigma' \gtrsim \frac{B' \log^2 n}{\sqrt{\ell_\flat}}$. Hence we can apply part 2 of Lemma 17 to obtain

$$\|\mathcal{P}_\flat Q_\not\sim\| \le 10\sigma'\sqrt{\ell_\flat} \le \frac{1}{4}, \text{w.h.p.},$$

where in the last inequality we use $n \ge \ell_\flat$ and $t(1-t) \ge \frac{1}{16}p(1-q) \gtrsim \frac{\log^4 n}{n}$.

(c) Note that $\mathcal{P}_\sharp Q_\sim = \mathcal{P}_\sharp Q_1 + \mathcal{P}_\sharp Q_2$. By construction these two matrices are both block-diagonal, have independent zero-mean entries which are bounded almost surely by $B_{\sim,1} := \frac{1}{\ell_\sharp p}$ and $B_{\sim,2} := \frac{2c_2}{p}$ respectively, and and have variance bounded by $\sigma^2_{\sim_1} := \frac{1}{p\ell_\sharp^2}$ and $\sigma^2_{\sim_2} := \frac{4(1-t)}{p}c_2^2$ respectively. One verifies that $\sigma_{\sim,i} \gtrsim \frac{B_{\sim,i}\log^2 n}{\sqrt{n}}$ for $i = 1, 2$. We can then apply part 2 of Lemma 17 to obtain $\|\mathcal{P}_\sharp Q_\sim\| \le 10(\sigma_{\sim,1} + \sigma_{\sim,2})\sqrt{n} \le \frac{1}{4}$ w.h.p.

(d) Note that $\mathcal{P}_\sharp Q_\not\sim = \mathcal{P}_\sharp Q_3$ is a random matrix with independent zero-mean entries which are bounded almost surely by $B_\not\sim := \frac{2c_1}{1-q}$ and have variance bounded by $\sigma^2_\not\sim := \frac{4t}{1-q}c_1^2$. One verifies that $\sigma_\not\sim \ge \frac{B_\not\sim \log^2 n}{\sqrt{n}}$. We can then apply part 2 of Lemma 17 to obtain $\|\mathcal{P}_\sharp Q_\not\sim\| \le 4\sigma_\not\sim\sqrt{n} \le \frac{1}{4}$ w.h.p.

Property 2:

Due to the structure of T, we have

$$\begin{aligned}\|\mathcal{P}_T Q\|_\infty = \|\mathcal{P}_T\mathcal{P}_\sharp Q\|_\infty &= \left\|UU^\top(\mathcal{P}_\sharp Q) + (\mathcal{P}_\sharp Q)UU^\top + UU^\top(\mathcal{P}_\sharp Q)UU^\top\right\|_\infty \\ &\le 3\left\|UU^\top\mathcal{P}_\sharp Q\right\|_\infty \le 3\sum_{m=1}^{3}\left\|UU^\top\mathcal{P}_\sharp Q_m\right\|_\infty.\end{aligned}$$

Now observe that $(UU^\top\mathcal{P}_\sharp Q_m)(i,j) = \sum_{l\in V_{\langle i\rangle}}\frac{1}{n_{\langle i\rangle}}\mathcal{P}_\sharp Q_m(l,j)$ is the sum of independent zero-mean random variables with bounded magnitude and variance. Using the Bernstein inequality in Lemma 19, we obtain that for each (i,j) and with probability at least $1-n^{-8}$,

$$\left|(UU^\top\mathcal{P}_\sharp Q_1)(i,j)\right| \le \frac{10}{n_{\langle i\rangle}\ell_\sharp}\left(\sqrt{\frac{1-p}{p}}\cdot\sqrt{n_{\langle i\rangle}\log n} + \frac{\log n}{p}\right) \le \frac{1}{24\kappa}\sqrt{\frac{\log^2 n}{n\ell_\sharp}}, \quad \text{w.h.p.},$$

where in the last inequality we use $p \gtrsim \frac{\kappa^2 n}{\ell_\sharp^2}$. For $i \in V_\flat$, clearly $(UU^\top\mathcal{P}_\sharp Q_1)(i,j) = 0$. By union bound we conclude that $\left\|UU^\top\mathcal{P}_\sharp Q_1\right\|_\infty \le \frac{1}{24\kappa}\sqrt{\frac{\log^2 n}{n\ell_\sharp}}$ w.h.p. We can bound $\left\|UU^\top\mathcal{P}_\sharp Q_2\right\|_\infty$ and $\left\|UU^\top\mathcal{P}_\sharp Q_3\right\|_\infty$ in a similar fashion: for each (i,j) and with probability at least $1-n^{-8}$:

$$\begin{aligned}\left|(UU^\top\mathcal{P}_\sharp Q_2)(i,j)\right| &\le 10\frac{(1+\epsilon)c_2}{n_{\langle i\rangle}}\left(\sqrt{\frac{1-p}{p}}\cdot\sqrt{n_{\langle i\rangle}\log n} + \frac{\log n}{p}\right) \\ &\le \frac{15}{100\kappa}\sqrt{\frac{t}{(1-t)n}}\cdot\left(\sqrt{\frac{(1-p)\log n}{p\ell_\sharp}} + \frac{\log n}{\ell_\sharp p}\right) \le \frac{1}{6\kappa}\sqrt{\frac{\log^2 n}{n\ell_\sharp}},\end{aligned}$$

where the last inequality follows from $p(1-t) \gtrsim \frac{\log n}{\ell_\sharp}$, and

$$\begin{aligned}\left|(UU^\top \mathcal{P}_\sharp Q_3)(i,j)\right| &\le 10\frac{(1+\epsilon)c_1}{n_{\langle i\rangle}}\left(\sqrt{\frac{q}{1-q}}\cdot\sqrt{n_{\langle i\rangle}\log n}+\frac{\log n}{1-q}\right)\\ &\le \frac{15}{100\kappa}\sqrt{\frac{1-t}{tn}}\cdot\left(\sqrt{\frac{q\log n}{(1-q)\ell_\sharp}}+\frac{\log n}{\ell_\sharp(1-q)}\right)\le \frac{1}{6\kappa}\sqrt{\frac{\log^2 n}{n\ell_\sharp}},\end{aligned}$$

where the last inequality follows from $t(1-q) \gtrsim \frac{\log n}{\ell_\sharp}$. On the other hand, under the definition of c_1, c_2 and ϵ, we have

$$c_1\epsilon \ge \frac{1}{100\kappa}\sqrt{\frac{1-t}{tn}}\cdot 100\sqrt{\frac{\log^4 n}{t(1-t)\ell_\sharp}} = \frac{1}{\kappa t}\cdot\sqrt{\frac{\log^4 n}{n\ell_\sharp}} \ge \frac{3}{\kappa}\sqrt{\frac{\log^2 n}{n\ell_\sharp}},$$

and similarly

$$c_2\epsilon \ge \frac{1}{100\kappa}\sqrt{\frac{t}{(1-t)n}}\cdot 100\sqrt{\frac{\log^4 n}{t(1-t)\ell_\sharp}} \ge \frac{3}{\kappa}\sqrt{\frac{\log^2 n}{n\ell_\sharp}}.$$

It follows that $\|\mathcal{P}_T Q\|_\infty \le 3\cdot\left(\frac{1}{24}+\frac{1}{6}+\frac{1}{6}\right)\cdot\frac{\epsilon}{3}(c_1\wedge c_2) \le \frac{\epsilon}{2}(c_1\wedge c_2)$ w.h.p., proving Property 2).

Properties 3(a) and 3(b):

For 3(a), by construction of Q we have

$$\begin{aligned}\left\langle UU^\top + Q, \mathcal{P}_{\mathrm{s}(A)}\mathcal{P}_{\mathrm{s}(\hat{B})}\mathcal{P}_\sharp\Delta\right\rangle &= \left\langle \mathcal{P}_{\mathrm{s}(A)}\mathcal{P}_{\mathrm{s}(\hat{B})}\mathcal{P}_\sharp Q_3, \mathcal{P}_{\mathrm{s}(A)}\mathcal{P}_{\mathrm{s}(\hat{B})}\mathcal{P}_\sharp\Delta\right\rangle\\ &= (1+\epsilon)c_1\cdot\sum_{(i,j)\in\mathrm{s}(\hat{B})\cap\mathrm{s}(A)}\mathcal{P}_\sharp\Delta(i,j)\\ &= (1+\epsilon)c_1\left\|\mathcal{P}_{\mathrm{s}(A)}\mathcal{P}_{\mathrm{s}(\hat{B})}\mathcal{P}_\sharp\Delta\right\|_1,\end{aligned}$$

where the last equality follows from $\Delta \in \mathfrak{D}$. Similarly, since

$$\mathcal{P}_{\mathrm{s}(A)^c}\mathcal{P}_{\mathrm{s}(\hat{B})}\mathcal{P}_\sharp Q_1 = \mathcal{P}_{\mathrm{s}(A)^c}\mathcal{P}_{\mathrm{s}(\hat{B})}\mathcal{P}_\sharp(-UU^\top),$$

we have

$$\begin{aligned}\left\langle UU^\top + Q, \mathcal{P}_{\mathrm{s}(A)^c}\mathcal{P}_{\mathrm{s}(\hat{B})}\mathcal{P}_\sharp\Delta\right\rangle &= \left\langle \mathcal{P}_{\mathrm{s}(A)^c}\mathcal{P}_{\mathrm{s}(\hat{B})}\mathcal{P}_\sharp Q_2, \mathcal{P}_{\mathrm{s}(A)^c}\mathcal{P}_{\mathrm{s}(\hat{B})}\mathcal{P}_\sharp\Delta\right\rangle\\ &= -(1+\epsilon)c_2\cdot\sum_{(i,j)\in\mathrm{s}(\hat{B})\cap\mathrm{s}(A)^c}\mathcal{P}_\sharp\Delta(i,j)\\ &= (1+\epsilon)c_2\left\|\mathcal{P}_{\mathrm{s}(A)^c}\mathcal{P}_{\mathrm{s}(\hat{B})}\mathcal{P}_\sharp\Delta\right\|_1,\end{aligned}$$

where the last equality again follows from $\Delta \in \mathfrak{D}$; this proves Property 3(b).

Properties 4(a) and 4(b):

For 4(a), we have

$$\begin{aligned}
&\left\langle UU^\top + Q, \mathcal{P}_{\mathrm{s}(A)}\mathcal{P}_{\mathrm{s}(\hat{B})^c}\mathcal{P}_\sharp\Delta\right\rangle \\
=&\left\langle \mathcal{P}_{\mathrm{s}(A)}\mathcal{P}_{\mathrm{s}(\hat{B})^c}\mathcal{P}_\sharp\left(UU^\top + \mathcal{P}_\sharp Q_1 + \mathcal{P}_\sharp Q_2\right), \mathcal{P}_{\mathrm{s}(A)}\mathcal{P}_{\mathrm{s}(\hat{B})^c}\mathcal{P}_\sharp\Delta\right\rangle \\
=&\sum_{(i,j)\in \mathrm{s}(\hat{B})^c\cap \mathrm{s}(A)} \left(\frac{1}{n_{\langle i\rangle}} + \frac{1}{n_{\langle i\rangle}}\frac{1-p_{ij}}{p_{ij}} + (1+\epsilon)c_2\frac{1-p_{ij}}{p_{ij}}\right)\mathcal{P}_\sharp\Delta(i,j) \\
\geq& -\left(\frac{1}{p\ell_\sharp} + (1+\epsilon)c_2\frac{1-p}{p}\right)\left\|\mathcal{P}_{\mathrm{s}(A)}P_{\mathrm{s}(\hat{B})^c}\mathcal{P}_\sharp\Delta\right\|_1, \qquad (12)
\end{aligned}$$

where the last inequality follows from $\Delta \in \mathfrak{D}, p_{ij} \geq p$ and $n_{\langle i\rangle} \geq \ell_\sharp, \forall i \in V_\sharp$. Consider the two terms in the parenthesis in (12). For the first term, we have

$$\frac{1}{p\ell_\sharp} = \frac{100\kappa}{\ell_\sharp}\sqrt{\frac{n}{t(1-t)}}\cdot\sqrt{\frac{t(1-t)}{100^2\kappa^2p^2n}} \leq \frac{100\kappa}{\ell_\sharp}\sqrt{\frac{n}{t(1-t)}}\cdot\frac{1}{100\kappa}\sqrt{\frac{1-t}{tn}} \leq \epsilon c_1.$$

For the second term in (12), we have the following:

$$\begin{aligned}
p - t \geq \frac{p-q}{4} \geq& \frac{1}{4}\max\left\{\frac{\kappa\sqrt{b_3p(1-q)n}}{\ell_\sharp}, \sqrt{\frac{b_3p(1-q)\log^4 n}{\ell_\sharp}}\right\} \\
=&\frac{\sqrt{b_3}}{4}\cdot p(1-t)\cdot\frac{\sqrt{t(1-q)}}{\sqrt{p(1-t)}}\cdot\max\left\{\frac{\kappa\sqrt{n}}{\ell_\sharp\sqrt{t(1-t)}}, \sqrt{\frac{\log^4 n}{t(1-t)\ell_\sharp}}\right\} \\
\geq& 8p(1-t)\cdot 100\max\left\{\frac{\kappa\sqrt{n}}{\ell_\sharp\sqrt{t(1-t)}}, \sqrt{\frac{\log^4 n}{t(1-t)\ell_\sharp}}\right\} = 8p(1-t)\epsilon.
\end{aligned}$$

A little algebra shows that this implies $(1+\epsilon)\sqrt{\frac{t}{1-t}}\frac{1-p}{p} \leq (1-\epsilon)\sqrt{\frac{1-t}{t}}$, or equivalently $(1+\epsilon)c_2\frac{1-p}{p} \leq (1-2\epsilon)c_1$. Substituting back to (12), we conclude that

$$\left\langle UU^\top + Q, \mathcal{P}_{\mathrm{s}(A)}\mathcal{P}_{\mathrm{s}(\hat{B})^c}\mathcal{P}_\sharp\Delta\right\rangle \geq -\left(\epsilon c_1 + (1-2\epsilon)c_1\right)\left\|\mathcal{P}_{\mathrm{s}(A)}P_{\mathrm{s}(\hat{B})^c}\mathcal{P}_\sharp\Delta\right\|_1,$$

proving Property 4(a).

For 4(b), we have

$$\begin{aligned}
\left\langle UU^\top + Q, \mathcal{P}_{\mathrm{s}(A)^c}\mathcal{P}_{\mathrm{s}(\hat{B})^c}\mathcal{P}_\sharp\Delta\right\rangle &= \left\langle \mathcal{P}_{\mathrm{s}(A)^c}\mathcal{P}_{\mathrm{s}(\hat{B})^c}\mathcal{P}_\sharp Q_3, \mathcal{P}_{\mathrm{s}(A)^c}\mathcal{P}_{\mathrm{s}(\hat{B})^c}\mathcal{P}_\sharp\Delta\right\rangle \\
&= \sum_{(i,j)\in \mathrm{s}(A)^c\cap \mathrm{s}(\hat{B})^c} -(1+\epsilon)\frac{c_1q_{ij}}{1-q_{ij}}\mathcal{P}_\sharp\Delta(i,j) \\
&\geq -(1+\epsilon)\frac{c_1q}{1-q}\left\|\mathcal{P}_{\mathrm{s}(A)^c}\mathcal{P}_{\mathrm{s}(\hat{B})^c}\mathcal{P}_\sharp\Delta\right\|_1, \qquad (13)
\end{aligned}$$

where the last inequality follows from $q_{ij} \le q$. Consider the factor before the norm in (13). Similarly as before, we have

$$t - q \ge \frac{p-q}{4} \ge \frac{1}{4}\max\left\{\frac{\kappa\sqrt{b_3 p(1-q)n}}{\ell_\sharp}, \sqrt{\frac{b_3 p(1-q)\log^4 n}{\ell_\sharp}}\right\}$$
$$\ge 2t(1-q)\cdot 100\max\left\{\frac{\kappa\sqrt{n}}{\ell_\sharp\sqrt{t(1-t)}}, \sqrt{\frac{\log^4 n}{t(1-t)\ell_\sharp}}\right\} = 2t(1-q)\epsilon.$$

A little algebra shows that this implies $(1+\epsilon)\sqrt{\frac{1-t}{t}\frac{q}{1-q}} \le (1-\epsilon)\sqrt{\frac{t}{1-t}}$, or equivalently $(1+\epsilon)c_1\frac{q}{1-q} \le (1-\epsilon)c_2$. Substituting back to (13), we conclude that

$$\left\langle UU^\top + Q, \mathcal{P}_{\mathrm{s}(A)^c}\mathcal{P}_{\mathrm{s}(\hat{B})^c}\mathcal{P}_\sharp\Delta\right\rangle \ge -(1-\epsilon)c_2\left\|\mathcal{P}_{\mathrm{s}(A)^c}\mathcal{P}_{\mathrm{s}(\hat{B})^c}\mathcal{P}_\sharp\Delta\right\|_1,$$

proving Property 4(b).

Properties 5 and 6:

Note that $\mathcal{P}_\flat UU^\top = 0$ and $\mathcal{P}_{\mathrm{s}(\hat{B})}\mathcal{P}_\flat = \mathcal{P}_{\mathrm{s}(A)}\mathcal{P}_\flat$. These two properties hold by construction of Q.

We note that Properties (3)-(6) hold deterministically.

Combining the above results and applying the union bound, we conclude that with probability at least $1 - n^{-3}$, there exists a matrix Q (which is the one constructed and verified above) that satisfies the properties in Proposition 12, where the probability is with respect to the randomness in the graph A and the matrix W. Since W is independent of A, integrating out the randomness in W proves the theorem.

5.2 Proof of Theorem 5

To ease notation, throughout the proof, C denotes a general universal positive constant that can take different values at different locations. We let $\Omega := \mathrm{s}(B^*)$ denote the *noise locations*.

Fix $\kappa \ge 1$ and t in the allowed range, let (K, B) be an optimal solution to (CP), and assume K is a partial clustering induced by $U_1, \ldots, U_r$ for some integer r, and also assume $\sigma_{\min}(K) = \min_{i\in[r]}|U_i|$ satisfies (5). Let $M = \sigma_{\min}(K)$. We need a few helpful facts. Note that from the definition of t, c_1, c_2,

$$q + \frac{1}{4}(p-q) \le \frac{c_2}{c_1+c_2} = t \le p - \frac{1}{4}(p-q)\ . \tag{14}$$

We say that a pair of sets $Y \subseteq V, Z \subseteq V$ is *cluster separated* if there is no pair $(y,z) \in Y \times Z$ satisfying $y \sim z$.

Assumption 13 *For all pairs of cluster-separated sets Y, Z of size at least $m := \frac{C\log n}{(p-q)^2}$ each,*

$$|\hat{d}_{Y,Z} - q| < \frac{1}{4}(p-q)\ , \tag{15}$$

where $\hat{d}_{Y,Z} := \frac{|(Y\times Z)\cap\Omega|}{|Y|\cdot|Z|}$.

This is proven by a Hoeffding tail bound and a union bound to hold with probability at least $1 - n^{-4}$. To see why, fix the sizes m_Y, m_Z of $|Y|, |Z|$, assume $m_Y \leq m_Z$ w.l.o.g. For each such choice, there are at most $\exp\{C(m_Y + m_Z)\log n\} \leq \exp\{2Cm_Z \log n\}$ possibilities for the choice of sets Y, Z. For each such choice, the probability that (15) does not hold is

$$\exp\{-Cm_Y m_Z (p-q)^2\} \tag{16}$$

using Hoeffding inequality. Hence, as long as $m_Y \geq m$ as defined above, using union bound (over all possibilities of m_Y, m_Z and of Y, Z) we obtain (15) uniformly. If we also assume that

$$M \geq 3m \ , \tag{17}$$

the implication of Assumption 13 is that it cannot be the case that some U_i contains a subset U_i' of size in the range $[m, |U_i| - m]$ such that $U_i' = V_g \cap U_i$ for some g. Otherwise, if such a set existed, then we would find a strictly better solution to (CP), call it (K', B'), which is defined so that K' is obtained from K by splitting the block corresponding to U_i into two blocks, one corresponding to U_i' and the other to $U_i \setminus U_i'$. The difference Δ between the cost of (K, B) and (K', B') is (renaming $Y := U_i'$ and $Z := U \setminus U_i'$)

$$\Delta = c_1|(Y \times Z) \cap \Omega| - c_2|(Y \times Z) \cap \Omega^c| = (c_1 + c_2)\hat{d}_{Y,Z}|Y|\,|Z| - c_2|Y|\,|Z| \ .$$

But the sign of Δ is exactly the sign of $\hat{d}_{Y,Z} - \frac{c_2}{c_1+c_2}$ which is strictly negative by (15) and (14). (We also used the fact that the trace norm part of the utility function is equal for both solutions: $\|K'\|_* = \|K\|_*$).

The conclusion is that for each i, the sets $(U_i \cap V_1), \ldots, (U_i \cap V_k)$ must all be of size at most m, except maybe for at most one set of size at least $|U_i| - m$. But note that by the theorem's assumption,

$$M > km = (kC \log n)/(p-q)^2 \ , \tag{18}$$

so we conclude that not all the sets $(U_i \cap V_1), \ldots, (U_i \cap V_k)$ can be of size at most m. Hence exactly one of these sets must have size at least $|U_i| - m$. From this we conclude that there is a function $\phi : [r] \mapsto [k]$ such that for all $i \in [r]$,

$$|U_i \cap V_{\phi(i)}| \geq |U_i| - m \ .$$

We now claim that this function is an injection. We will need the following assumption:

Assumption 14 *For any 4 pairwise disjoint subsets (Y, Y', Z, Z') such that $(Y \cup Y') \subseteq V_i$ for some i, $(Z \cup Z') \subseteq [n] \setminus V_i$, $\max\{|Z|, |Z'|\} \leq m$, $\min\{|Y|, |Y'|\} \geq M - m$:*

$$\begin{aligned} |Y| \cdot |Y'|\,\hat{d}_{Y,Y'} - |Y| \cdot |Z|\,\hat{d}_{Y,Z} - |Y'| \cdot |Z'|\,\hat{d}_{Y',Z'} > \\ \frac{c_2}{c_1 + c_2}(|Y| \cdot |Y'| - |Y| \cdot |Z| - |Y'| \cdot |Z'|) \end{aligned} \tag{19}$$

The assumption holds with probability at least $1 - n^{-4}$ by using Hoeffding inequality, union bounding over all possible sets Y, Y', Z, Z' as above. Indeed, notice that for fixed $m_Y, m_{Y'}, m_Z, m_{Z'}$ (with, say, $m_Y \geq m_{Y'}$), and for each tuple Y, Y', Z, Z' such that $|Y| = m_Y, |Y'| = m_{Y'}, |Z| = m_Z, |Z'| = m_{Z'}$, the probability that (19) is violated is at most

$$\exp\{-C(p-q)^2(m_Y m_{Y'} + m_Y m_Z + m_{Y'} m_{Z'})\} \ . \tag{20}$$

Using (17), this is at most

$$\exp\{-C(p-q)^2(m_Y m_{Y'})\} \ . \tag{21}$$

Now notice that the number of possibilities to choose such a 4 tuple of sets is bounded above by $\exp\{Cm_Y \log n\}$. Assuming

$$M \geq \frac{C\log n}{(p-q)^2} \ , \tag{22}$$

and applying a union bound over all possible combinations Y, Y', Z, Z' of sizes $m_Y, m_{Y'}, m_Z, m_{Z'}$ respectively, of which there are at most $\exp\{Cm_Y \log n\}$, we conclude that (19) is violated for some combination with probability at most

$$\exp\{-C(p-q)^2 m_Y m_{Y'}/2\} \tag{23}$$

which is at most $\exp\{-C\log n\}$ if

$$M \geq \frac{C\log n}{(p-q)^2} \ . \tag{24}$$

Apply a union bound now over the possible combinations of the tuple $(m_Y, m_{Y'}, m_Z, m_{Z'})$, of which there are at most $\exp\{C\log n\}$ to conclude that (19) holds uniformly for all possibilities of Y, Y', Z, Z' with probability at least $1 - n^{-4}$.

Now assume by contradiction that ϕ is not an injection, so $\phi(i) = \phi(i') =: j$ for some distinct $i, i' \in [r]$. Set $Y = U_i \cap V_j, Y' = U_{i'} \cap V_j$, $Z = U_i \setminus Y, Z' = U_{i'} \setminus Y'$. Note that $\max\{|Z|, |Z'|\} \leq m$ and $\min\{|Y|, |Y'|\} \geq M - m$ by the derivations to this point. Consider the solution (K', B') where K' is obtained from K by replacing the two blocks corresponding to $U_i, U_{i'}$ with four blocks: Y, Y', Z, Z'. Inequality (19) guarantees that the cost of (K', B') is strictly lower than that of (K, B), contradicting optimality of the latter. (Note that we used the fact that the corresponding contributions $\|K\|_*$ and $\|K'\|_*$ to the trace-norm part of the utility function are equal.)

We can now also conclude that $r \leq k$. Fix $i \in [r]$. We show that not too many elements of $V_{\phi(i)}$ can be contained in $V \setminus \{U_1 \cup \cdots \cup U_r\}$. We need the following assumption.

Assumption 15 *For all pairwise disjoint sets $Y, X, Z \subseteq V$ such that $|Y| \geq M - m$, $|X| \geq m$, $(Y \cup X) \subseteq V_j$ for some $j \in [k]$, $|Z| \leq m$, $Z \cap V_j = \emptyset$:*

$$\begin{aligned} |X|\cdot|Y|\hat{d}_{X,Y} + \binom{|X|}{2}\hat{d}_{x,x} - |Y|\cdot|Z|\hat{d}_{Y,Z} > \\ \frac{c_2}{c_1+c_2}(|X|\cdot|Y| + \binom{|X|}{2} - |Y|\cdot|Z|) + \frac{|X|}{c_1+c_2} \ . \end{aligned} \tag{25}$$

The assumption holds with probability at least $1 - n^{-4}$. To see why, first notice that $|X|/(c_1+c_2) \leq \frac{1}{8}(p-q)|X|\cdot|Y|$ by (5), as long as C_2 is large enough. This implies that the RHS of (25) is upper bounded by

$$\left(p - \frac{1}{8}(p-q)\right)|X|\cdot|Y| + \frac{c_2}{c_1+c}\left(\binom{|X|}{2} - |Y|\cdot|Z|\right) \tag{26}$$

Proving that the LHS of (25) (denoted $f(X,Y,Z)$) is larger than (26) (denoted $g(X,Y,Z)$) uniformly w.h.p. can now be easily done as follows. By fixing $m_Y = |Y|, m_X = |X|$, the number of combinations for Y, X, Z is at most $\exp\{C(m_Y + m_X)\log n\}$ for some global $C > 0$. On the other hand, the probability that $f(X,Y,Z) \leq g(X,Y,Z)$ for any such option is at most

$$\exp\{-C(p-q)^2 m_Y m_X\} \ . \tag{27}$$

Hence, by union bounding, the probability that some tuple Y, X, Z of sizes m_Y, m_X, m_Z respectively satisfies $f(X,Y,Z) \leq g(X,Y,Z)$ is at most

$$\exp\{-C(p-q)^2 m_Y/2\} \ , \tag{28}$$

which is at most $\exp\{-C\log n\}$ assuming

$$M \geq C(\log n)/(p-q)^2 \ . \tag{29}$$

Another union bound over the possible choices of m_Y, m_X, m_Z proves that (25) holds uniformly with probability at least $1 - n^{-4}$.

Now assume, by way of contradiction, that for some $i \in [r]$, the set $X := V_{\phi(i)} \cap ([n] \setminus \{U_1 \cup \cdots \cup U_r\})$ is of size greater than m. Set $Y := V_{\phi(i)} \cap U_i$ and $Z = U_i \setminus V_{\phi(i)}$. Define the solution (K', B') where K' is obtained from K by replacing the block corresponding to $U_i = Y \cup Z$ in K with two blocks: $Y \cup X$ and Z. Assumption 15 tells us that the cost of (K', B') is strictly lower than that of (K, B). Note that the expression $\frac{|X|}{c_1+c_2}$ in the RHS of (25) accounts for the trace norm difference $\|K'\|_* - \|K\|_* = |X|$.

We are prepared to perform the final "cleanup" step. At this point we know that for each $i \in [r]$, the set $T_i = U_i \cap V_{\phi(i)}$ satisfies

$$t_i := |T_i| \geq \max\{|U_i| - m, |V_{\phi(i)}| - rm\} \ . \tag{30}$$

(To see why $t_i \geq |V_{\phi(i)}| - rm$, note that at most m elements of $V_{\phi(i)}$ may be contained in $U_{i'}$ for $i' \neq i$, and another at most m elements in $V \setminus (U_1 \cup \cdots \cup U_r)$.) We are now going to conclude from this that $U_i = V_{\phi(i)}$ for all i. To that end, let (K', B') be the feasible solution to (CP) defined so that K' is a partial clustering induced by $V_{\phi(1)}, \ldots, V_{\phi(r)}$. We would like to argue that if $K \neq K'$ then the cost of (K', B') is strictly smaller than that of (K, B). Fix the value of the collection

$$\begin{aligned}\mathcal{Y} := ((&r, \phi(1), \ldots, \phi(r),\\ &\big(m_{ij} := |V_{\phi(i)} \cap U_j|\big)\big)_{i,j\in[r],i\neq j} \ ,\\ &\big(m'_i := |V_{\phi(i)} \cap (V \setminus (U_1 \cup \cdots \cup U_r))|\big)_{i\in[r]}).\end{aligned}$$

Let $\beta(\mathcal{Y})$ denote the number of $i \neq j$ such that $m_{ij} > 0$ plus the number of $i \in [r]$ such that $m'_i > 0$. We can assume $\beta(\mathcal{Y}) > 0$, otherwise $U_i = V_{\phi(i)}$ for all $i \in [r]$. The number of possibilities for K giving rise to $\mathcal{Y}$ is $\exp\{C(\sum_{i\neq j} m_{ij} + \sum_i m_i)\log n\}$. Fix such a possibility, and let

$$D_{ij} = V_{\phi(i)} \cap U_j, \qquad D'_i = V_{\phi(i)} \cap (V \setminus (U_1 \cup \cdots \cup U_r)) \ .$$

The difference $\delta(K, K')$ between the (CP) costs of solutions K and K' is given by the following expression:

$$\begin{aligned}\delta =& c_1 \sum_i \sum_{j\neq i} |(D_{ij} \times U_i) \cap \Omega| + c_1 \sum_i \sum_{\substack{j_1<j_2 \\ j_1,j_2\neq i}} |(D_{ij_1} \times D_{ij_2}) \cap \Omega| \\ &+ c_1 \sum_i |((V_{\phi(i)} \setminus D'_i) \times D'_i) \cap \Omega| + c_2 \sum_i \sum_{j\neq i} |(D_{ij} \times U_j) \cap \Omega^c| \\ &- c_2 \sum_i \sum_{j\neq i} |(D_{ij} \times U_i) \cap \Omega^c| - c_2 \sum_i \sum_{\substack{j_1<j_2 \\ j_1,j_2\neq i}} |(D_{ij_1} \times D_{ij_2}) \cap \Omega^c| \\ &- c_2 \sum_i |((V_{\phi(i)} \setminus D'_i) \times D'_i) \cap \Omega^c| - c_1 \sum_i \sum_{j\neq i} |(D_{ij} \times U_j) \cap \Omega| - \sum m'_i \ ,\end{aligned}$$

where the expression $\sum m'_i$ comes from the trace norm contribution. If the quantity $\delta(K, K')$ is non-positive, then at least one of the following must be true:

(i) $\sum_i \sum_{j\neq i} |(D_{ij} \times U_i) \cap \Omega| + \sum_i \sum_{\substack{j_1<j_2 \\ j_1,j_2\neq i}} |(D_{ij_1} \times D_{ij_2}) \cap \Omega|$
$< \frac{c_2}{c_1+c_2} \sum_i \sum_{j\neq i} |(D_{ij} \times U_i)| + \frac{c_2}{c_1+c_2} \sum_i \sum_{\substack{j_1<j_2 \\ j_1,j_2\neq i}} |(D_{ij_1} \times D_{ij_2})|$

(ii) $\sum_i |((V_{\phi(i)} \setminus D'_i) \times D'_i) \cap \Omega| < \frac{c_1}{c_1+c_2} \sum_i |((V_{\phi(i)} \setminus D'_i) \times D'_i)| + \frac{1}{c_1+c_2} \sum_i m'_i$.

(iii) $\sum_i \sum_{j\neq i} |(D_{ij} \times U_j) \cap \Omega| > \frac{c_2}{c_1+c_2} \sum_i \sum_{j\neq i} |(D_{ij} \times U_j)|$.

Inequality (i) occurs with probability at most

$$\exp\left\{-C(p-q)^2 \sum_i M \sum_{j\neq i} m_{ij}\right\} \tag{31}$$

using Hoeffding bound; we also used (30). Inequality (ii) occurs with probability at most

$$\exp\left\{-C(p-q)^2 \sum_i M m'_i\right\} \tag{32}$$

using Hoeffding inequalities. (We also used the fact that the rightmost expression of (ii), $\frac{1}{c_1+c_2} \sum_i m'_i$, is bounded above by $\frac{1}{4}(p-q) \sum_i |((V_{\phi(i)} \setminus D'_i) \times D'_i)|$ due to the theorem assumptions.) Inequality (iii) occurs occurs with probability at most (31), using Hoeffding bounds again.

Now notice that the number of choices of K' giving rise to our fixed $\mathcal{Y}$ and $\beta(\mathcal{Y})$ is, by a gross estimation, at most $\exp\{(\sum_{j\neq i} m_{ij} + \sum_i m'_i) \log n\}$. The assumptions of the theorem ensure that, using a union bound over all such possibilities K, and then over all options for $\beta(\mathcal{Y})$ and $\mathcal{Y}$, with probability at least $1 - n^{-4}$ the difference $\delta(K, K')$ is positive. This means that the (CP) cost of K' is simultaneously strictly lower than that of K for all K we have enumerated over.

Taking the theorem's C_1, C_2 large enough to satisfy the requirements above concludes the proof.

5.3 Proof of Theorem 9

The proof of Theorem 5 in the previous section made repeated use of Hoeffding tail inequalities, for uniformly bounding the size of the intersection of the noise support Ω with various submatrices (with high probability). This is tight for p, q which are bounded away from 0 and 1. However, if $p = \rho p', q = \rho q'$, the noise probabilities p', q' are fixed and ρ tends to 0, a sharper bound is obtained using Bernstein tail bound (Lemma 18 in see Appendix A.2). Using Bernstein inequality instead of Hoeffding inequality, gives the required result. To see how this is done, the counterpart of Assumption 13 above is as follows:

Assumption 16 *For all pairs of cluster-separated sets Y, Z of size at least $m := \frac{C \log n}{\rho}$ each,*

$$|\hat{d}_{Y,Z} - q| < \frac{1}{4}(\rho p' - \rho q') \ , \tag{33}$$

where $\hat{d}_{Y,Z} := \frac{|(Y \times Z) \cap \Omega|}{|Y| \cdot |Z|}$.

Note: In this section, C (and hence also m) depends on p', q' only, which are assumed fixed. Defining henceforth m as in Assumption 9, Assumption 14 holds with probability at least $1 - n^{-4}$. This can be seen by replacing the Hoeffding bound in (20) with a Chernoff bound:

$$\exp\{-C(p', q')\rho(m_Y m_{Y'} + m_Y m_Z + m_{Y'} m_{Z'})\} \ . \tag{34}$$

The rest of the proof is obtained by a similar step by step technical alteration of the proof in Section 5.2.

6. Discussion

An immediate future research is to better understand the "mid-size crisis". Our current results say nothing about clusters that are neither large nor small, falling in the interval $(\ell_\flat, \ell_\sharp)$. Our numerical experiments confirm that the mid-size phenomenon is real: they are neither completely recovered nor entirely ignored by the optimal $\hat{K}$. The part of $\hat{K}$ restricted to these clusters does not seem to have an obvious pattern. Proving whether we can still efficiently recover large clusters in the presence of mid-size clusters is an interesting open problem.

Our study was mainly theoretical, focusing on the planted partition model. As such, our experiments focused on confirming the theoretical findings with data generated exactly according to the distribution we could provide provable guarantees for. It would be interesting to apply the presented methodology to real applications, particularly large data sets merged from web application and social networks.

Another interesting direction is extending the "peeling strategy" to other settings. Our algorithms use the convex program (CP) as a subroutine, taking advantage of the fact that the recovery of large clusters via (CP) is not hindered by the presence of small clusters, and that (CP) has a tunable parameter that controls the sizes of the clusters that are considered large. It is possible that other clustering routines also have these properties and thus can be used as a subroutine in our iterative and active algorithms. More generally, our problem concerns the inference of an unknown structure, and our high-level strategy is to sequentially infer and remove the "easy" (or low-resolution) part of the problem and

zoom into the "hard" (or high-resolution) part. It is interesting to explore this strategy in a broader context, and to understand for what problems and under what conditions this strategy may work.

Acknowledgments

The authors are grateful to the anonymous reviewers for their thorough reviews of this work and valuable suggestions on improving the manuscript. N. Ailon acknowledges the support of a Marie Curie International Reintegration Grant PIRG07-GA-2010-268403, and a grant from Technion-Cornell Innovation Institute (TCII). Y. Chen was supported by NSF grant CIF-31712-23800 and ONR MURI grant N00014-11-1-0688. The work of H. Xu was partially supported by the Ministry of Education of Singapore through AcRF Tier Two grant R265-000-443-112.

Appendix A. Technical Lemmas

In this section we state several lemmas needed in the proofs of our main results.

A.1 The Spectral Norm of Random Matrices

Lemma 17 *Suppose $A \in \mathbb{R}^{N\times N}$ is a symmetric matrix, where A_{ij}, $1 \le i \le j \le m$ are independent random variables, each of which has mean 0 and variance at most σ^2 and is bounded in absolute value by B a.s.*

1. *If $n \ge N$, then with probability at least $1 - n^{-6}$, the first singular value A satisfies*

$$\lambda_1(A) \le 10 \max\left\{\sigma\sqrt{N \log n}, B \log n\right\}.$$

2. *If further $n \ge N \ge 76$, $N \ge n^{2/3}$ and $\sigma \ge c_1 \frac{B \log^2 n}{\sqrt{N}}$ for some absolute constant $c_1 > 0$, then with probability at least $1 - n^{-6}$, we have*

$$\lambda_1(A) \le 10\sigma\sqrt{N}.$$

Proof We first prove part 1 of the lemma. Let e_i be the i-th standard basis in $\mathbb{R}^N$. Define $Z_{ij} = A_{ij}e_ie_j^\top + A_{ji}e_je_i^\top$ for $1 \le i < j \le N$, and $Z_{ii} = A_{ii}e_ie_i^\top$ for $i \in [N]$. Then the Z_{ij}'s are zero-mean random matrices independent of each other, and $A = \sum_{1\le i\le j\le N} Z_{ij}$. We have $\|Z_{ij}\| \le B$ almost surely. We also have

$$\left\|\sum_{1\le i\le j\le N} \mathbb{E}(Z_{ij}Z_{ij}^\top)\right\| = \left\|\sum_{1\le i\le N} \mathbb{E}(A_{ii}^2)e_ie_i^\top + \sum_{1\le i\le N} e_ie_i^\top \sum_{j:j\ne i} \mathbb{E}(A_{ij}^2)\right\| \le N\sigma^2.$$

Similarly, we have $\|\sum_{1\le i\le j\le N} \mathbb{E}(Z_{ij}^\top Z_{ij})\| \le N\sigma^2$. Applying the Matrix Bernstein Inequality (Theorem 1.6 in Tropp 2012) with $t = 10\max\left\{\sigma\sqrt{N\log n}, B\log n\right\}$ yields the desired bound.

We turn to part 2 of the lemma. Let A' be an independent copy of A, and define

$$\bar{A} := \begin{bmatrix} 0 & A \\ A' & 0 \end{bmatrix}.$$

Note that $\bar{A}$ is an $2N \times 2N$ random matrix with i.i.d. entries. If $\sigma \geq c_1 \frac{B \log^2 n}{\sqrt{N}}$ for some sufficiently large absolute constant $c_1 > 0$, then by Theorem 3.1 in Achlioptas and Mcsherry (2007) we know that with probability at least $1 - n^{-6}$, $\lambda_1(\bar{A}) \leq 10\sigma\sqrt{N}$. The lemma follows from noting that $\lambda_1(A) \leq \lambda_1(\bar{A})$. ■

A.2 Standard Bernstein Inequality for the Sum of Independent Variables

Lemma 18 *(*Bernstein inequality*) Let $Y_1, \ldots, Y_N$ be independent random variables, each of which has variance bounded by σ^2 and is bounded in absolute value by B a.s.. Then we have that*

$$\Pr\left[\left|\sum_{i=1}^{N} Y_i - \mathbb{E}\left[\sum_{i=1}^{N} Y_i\right]\right| > t\right] \leq 2\exp\left\{\frac{t^2/2}{N\sigma^2 + Bt/3}\right\}.$$

The following lemma is an immediate consequence of Lemma 18.

Lemma 19 *Let $Y_1, \ldots, Y_N$ be independent random variables, each of which has variance bounded by σ^2 and is bounded in absolute value by B a.s. Then we have*

$$\left|\sum_{i=1}^{N} Y_i - \mathbb{E}\left[\sum_{i=1}^{N} Y_i\right]\right| \leq 10\left(\sigma\sqrt{N\log n} + B\log n\right)$$

with probability at least $1 - 2n^{-8}$.

References

Dimitris Achlioptas and Frank Mcsherry. Fast computation of low-rank matrix approximations. *Journal of the ACM*, 54(2):9, 2007.

Nir Ailon, Yudong Chen, and Huan Xu. Breaking the small cluster barrier of graph clustering. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 995–1003, 2013.

Nir Ailon, Ron Begleiter, and Esther Ezra. Active learning using smooth relative regret approximations with applications. *Journal of Machine Learning Research*, 15:885–920, 2014.

Brendan P. W. Ames and Stephen A. Vavasis. Nuclear norm minimization for the planted clique and biclique problems. *Mathematical Programming*, 129(1):69–89, 2011.

Anima Anandkumar, Rong Ge, Daniel Hsu, and Sham M. Kakade. A tensor spectral approach to learning mixed membership community models. *Journal of Machine Learning Research*, 15:2239–2312, June 2014.

Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Machine Learning*, 56:89–113, 2004.

Béla Bollobás and Alex D. Scott. Max cut for random graphs with a planted partition. *Combinatorics, Probability and Computing*, 13(4-5):451–474, 2004.

Ravi B. Boppana. Eigenvalues and graph bisection: an average-case analysis. In *Proceedings of Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 280–285, 1987.

Emmanuel J. Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM*, 58:1–37, 2011.

Ted Carson and Russell Impagliazzo. Hill-climbing finds random planted bisections. In *Proceedings of the 12th Annual Symposium on Discrete Algorithms*, pages 903–909, 2001.

Venkat Chandrasekaran, Sujay Sanghavi, Pablo Parrilo, and Alan Willsky. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21(2):572–596, 2011.

Kamalika Chaudhuri, Fan Chung, and Alexander Tsiatas. Spectral clustering of graphs with general degrees in the extended planted partition model. In *Proceedings of the 25th Annual Conference on Learning Theory (COLT)*, pages 35.1–35.23, 2012.

Yudong Chen, Sujay Sanghavi, and Huan Xu. Clustering sparse graphs. In *Advances in Neural Information Processing Systems 25*, pages 2204–2212. Curran Associates, Inc., 2012.

Yudong Chen, Ali Jalali, Sujay Sanghavi, and Huan Xu. Clustering partially observed graphs via convex optimization. *Journal of Machine Learning Research*, 15:2213–2238, June 2014a.

Yudong Chen, Sujay Sanghavi, and Huan Xu. Improved graph clustering. *IEEE Transactions on Information Theory*, 60(10):6440–6455, 2014b.

Anne Condon and Richard M. Karp. Algorithms for graph partitioning on the planted partition model. *Random Structures and Algorithms*, 18(2):116–140, 2001.

Brian Eriksson, Gautam Dasarathy, Aarti Singh, and Robert Nowak. Active clustering: robust and efficient hierarchical clustering using adaptively selected similarities. In *Proceedings of International Conference on Artificial Intelligence and Statistics*, pages 260–268, 2011.

Martin Ester, Hans-Peter Kriegel, and Xiaowei Xu. A database interface for clustering in large spatial databases. In *Proceedings of 1st International Conference on Knowledge Discovery and Data Mining (KDD)*, 1995.

Ioannis Giotis and Venkatesan Guruswami. Correlation clustering with a fixed number of clusters. *Theory of Computing*, 2(1):249–266, 2006.

Paul W. Holland, Kathryn B. Laskey, and Samuel Leinhardt. Stochastic blockmodels: some first steps. *Social networks*, 5(2):109–137, 1983.

Ali Jalali, Yudong Chen, Sujay Sanghavi, and Huan Xu. Clustering partially observed graphs via convex optimization. In *Proceedigns of the 28th International Conference on Machine Learning*, pages 1001–1008, 2011.

Akshay Krishnamurthy and Aarti Singh. Low-rank matrix and tensor completion via adaptive sampling. In *Advances in Neural Information Processing Systems 26*, pages 836–844, 2013.

Akshay Krishnamurthy, Sivaraman Balakrishnan, Min Xu, and Aarti Singh. Efficient active algorithms for hierarchical clustering. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, pages 887–894, 2012.

Amit Kumar and Ravindran Kannan. Clustering with spectral norm and the k-means algorithm. In *Proceedings of 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 299–308, 2010.

Zhouchen Lin, Risheng Liu, and Zhixun Su. Linearized alternating direction method with adaptive penalty for low-rank representation. In *Advances in Neural Information Processing Systems 24*, pages 612–620, 2011.

Claire Mathieu and Warren Schudy. Correlation clustering with noisy input. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 712–728. SIAM, 2010.

Frank McSherry. Spectral partitioning of random graphs. In *Proceedings of 42nd IEEE Symposium on Foundations of Computer Science*, pages 529–537, 2001.

Nina Mishra, Robert Schreiber, Isabelle Stanton, and Robert E. Tarjan. Clustering social networks. In *Algorithms and Models for the Web-Graph*, pages 56–67. Springer, 2007.

Samet Oymak and Babak Hassibi. Finding dense clusters via low rank + sparse decomposition. arXiv:1104.5186v1, 2011.

Karl Rohe, Sourav Chatterjee, and Bin Yu. Spectral clustering and the high-dimensional stochastic block model. *Annals of Statistics*, 39:1878–1915, 2011.

Ohad Shamir and Naftali Tishby. Spectral clustering on a budget. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, pages 661–669, 2011.

Ron Shamir and Dekel Tsur. Improved algorithms for the random cluster graph model. *Random Structure and Algorithm*, 31(4):418–449, 2007.

Joel A. Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12(4):389–434, 2012.

Konstantin Voevodski, Maria-Florina Balcan, Heiko Röglin, Shang-Hua Teng, and Yu Xia. Active clustering of biological sequences. *Journal of Machine Learning Research*, 13: 203–225, 2012.

Huan Xu, Constantine Caramanis, and Sujay Sanghavi. Robust PCA via outlier pursuit. *IEEE Transactions on Information Theory*, 58(5):3047–3064, 2012.

Yahoo!-Inc. Graph partitioning. `http://research.yahoo.com/project/2368`, 2009.

Yunpeng Zhao, Elizaveta Levina, and Ji Zhu. Community extraction for social networks. *Proceedings of the National Academy of Sciences*, 108(18):7321–7326, 2011.

Journal of Machine Learning Research 16 (2015) 491-494 Submitted 11/11; Revised 7/14; Published 3/15

A Classification Module for Genetic Programming Algorithms in JCLEC

Alberto Cano ACANO@UCO.ES
José María Luna JMLUNA@UCO.ES
Amelia Zafra AZAFRA@UCO.ES
Sebastián Ventura SVENTURA@UCO.ES
Department of Computer Science and Numerical Analysis
Rabanales Campus, University of Córdoba, 14071, Córdoba, Spain

Editor: Mikio Braun

Abstract

JCLEC-Classification is a usable and extensible open source library for genetic programming classification algorithms. It houses implementations of rule-based methods for classification based on genetic programming, supporting multiple model representations and providing to users the tools to implement any classifier easily. The software is written in Java and it is available from `http://jclec.sourceforge.net/classification` under the GPL license.

Keywords: classification, evolutionary algorithms, genetic programming, JCLEC

1. Introduction

In the last decade, the increasing interest in storing information has led to its automatic processing, discovering knowledge that is potentially useful. Data mining involves the use of data analysis tools to discover this knowledge previously unknown, valid patterns, and close relationships in databases. One of the most used data mining tasks is classification, which learns from a set of training examples to produce predictions about future examples.

The classification models are being applied to enormous databases in areas such as bioinformatics, marketing, banks or web mining. Existing classification libraries provide algorithms following many different methodologies. However, it is difficult to find a library that contains GP (genetic programming) algorithms, an important evolutionary computation paradigm. The conceptual difficulty of GP makes it difficult to implement algorithms following this paradigm despite its algorithms perform well as it is proved by many researchers (Espejo et al., 2010).

GP is an efficient and flexible heuristic technique that uses complex representations such as trees. This technique provides comprehensible models, which are useful in different application domains. For instance, it is applied to supervised learning tasks like regression, classification and unsupervised learning tasks like clustering and association. In classification tasks, the application of GP is an important issue since it may offer results that are comprehensible to humans. Additionally, it offers interesting advantages such as flexibility, and the possibility of using different kinds of representations, e.g., decision trees, rule-based systems, discriminant functions, etc. An extension of GP is grammar-guided genetic pro-

gramming (G3P), which makes the knowledge extracted more expressive and flexible by means of a context-free grammar (McKay et al., 2010).

This paper presents an open source software for researchers and end-users to develop classification algorithms based on GP and G3P models. It is an intuitive and usable tool which extends the JCLEC evolutionary computation library (Ventura et al., 2007). The software presented includes some GP and G3P proposals described in literature, and provides the necessary classes and methods to develop any kind of evolutionary algorithms for solving classification problems easily.

This paper is organized as follows. Firstly, Section 2 provides a description of the module, its structure and the way to use it. Finally, the documentation and the requirements of this module are outlined in Section 3.

2. Description of the Module

The classification module is presented in this section, describing the library structure and its main characteristics.

2.1 Structure of the Module

The *net.sf.jclec.problem.classification.base* package roots the hierarchical structure of the classification module, and provides the abstract classes with the properties and methods that any classification algorithm must contain, e.g., *ClassificationAlgorithm*, *ClassificationReporter*, *Rule* and *RuleBase*. A new algorithm included in the module should inherit from these classes regardless the classification model. In this context, we focus on rule-based classifiers which comprise one or more classification rules, each of them being a knowledge representation model consisting of an antecedent and a consequent. The antecedent of each classification rule is made up of a series of conditions to be met by an instance to consider that it belongs to the class specified by the consequent.

Based on whether an algorithm uses a GP or G3P encoding, JCLEC-Classification makes a differentiation between expression-tree and syntax-tree respectively. In such a way, each GP classification individual is represented by means of the *ExprTreeRuleIndividual* class, which represents an individual, comprising all the features required to do it: the genotype, the phenotype and the fitness function value. The nodes and functions in GP trees are defined by the *ExprTreeSpecies* class. Similarly to GP individuals, the *SyntaxTreeRuleIndividual* class specifies all the features required to represent a G3P individual, while the *SyntaxTreeSpecies* allows us to define the terminal and nonterminal symbols of the grammar used to generate individuals. Furthermore, the module allows to encode multiple syntax and expression trees for Pittsburgh style encodings or multi expression programming by means of the *MultiExprTree* and *MultiSyntaxTree* classes.

In order to represent the phenotype of a rule-base individual, crisp and fuzzy rules are generated by using the *CrispRule* and *FuzzyRule* classes, respectively. These classes provide the antecedent of the rule in an expression-tree shape and the consequent assigned to this antecedent. In addition, methods to classify a whole data set or a particular instance are provided in these classes. These methods compute whether the antecedent of a rule satisfies an instance, returning the consequent of the rule, otherwise the instance is not covered by the antecedent and therefore no predictions can be made. Besides those packages that repre-

sent the main characteristics of any individual, the *net.sf.jclec.problem.classification.listener* package to make reports for the train and test classification processes is provided. This package contains the *RuleBaseReporter* class with methods to make reports specifying the classifier features such as the rule base, the number of rules, the average number of conditions, the percentage of correct predictions, the percentage of correct predictions per class, the geometric mean, the kappa rate and the confusion matrix.

Finally, it is noteworthy that several utility classes, which make it easy to load data from KEEL[1] and ARFF[2] formatted files, are provided by a *dataset* package. Three different attribute types may be represented by this package, integer, continuous and categorical, and a number of characteristics from the data set are given, comprising type of attributes, number of classes, number of instances, etc.

The module houses three G3P classification algorithms (De Falco et al., 2001; Bojarczuk et al., 2004; Tan et al., 2002), which can guide developers to write new algorithms.

2.2 Usage of the Module

Including new classification algorithms in this module is very simple. We focus on the algorithm described by Bojarczuk et al. (2004). This algorithm, which is provided in the module (see the *net.sf.jclec.problem.classification.algorithm.bojarczuk* package), is constructed with only three additional classes. One of them, the *BojarczukAlgorithm* class is inherited from the *ClassificationAlgorithm* class and provides the own features of this algorithm.

Another class required to be implemented is the evaluator, which computes the fitness of the individuals. This class, named *BojarczukEvaluator* in this algorithm, inherits from the JCLEC core *AbstractParallelEvaluator* class or from the *AbstractEvaluator* class, depending on whether the individuals are evaluated in a sequential or parallel way.

Finally, a class to define the grammar to be followed in the individual generation stage is implemented. This class, named *BojarczukSyntaxTreeSpecies* in this example, inherits from the class *SyntaxTreeSpecies* since G3P individuals are defined in this algorithm.

Only defining these three classes, the complete classification algorithm is represented. Due to the core of this module is JCLEC, before an algorithm is ready to run, it is necessary to carry out a set-up process by using a configuration file as shown in Figure 1. This configuration file and the steps required to execute the algorithm are described in the JCLEC website. In this file we specify those parameters required such as the algorithm to be run, the parent selector, the genetic operators, the evaluator, etc. All the required parameters are provided by JCLEC, existing a numerous variety of them as it is described in the JCLEC specification (Ventura et al., 2007).

3. Documentation and Requirements

The JCLEC-Classification online documentation[3] describes the software packages, presents a user oriented usage example, as well as developer information to include new algorithms, API reference and running tests. JCLEC requires Java 1.7, Apache commons logging 1.1,

1. KEEL website at `http://www.keel.es`
2. ARFF format description at `http://www.cs.waikato.ac.nz/ml/weka/arff.html`
3. JCLEC documentation at `http://jclec.sourceforge.net/data/JCLEC-classification.pdf`

```
<experiment>
   <process algorithm-type=''net.sf.jclec.problem.classification.bojarczuk.BojarczukAlgorithm''>
      <rand-gen-factory seed=''123456789'' type=''net.sf.jclec.util.random.RanecuFactory''/>
      <population-size>100</population-size>
      <max-of-generations>100</max-of-generations>
      <max-deriv-size>20</max-deriv-size>
      <dataset type=''net.sf.jclec.problem.util.dataset.ArffDataSet''>
         <train-data>data/iris/iris-10-1tra.arff</train-data>
          <test-data>data/iris/iris-10-1tst.arff</test-data>
         <attribute-class-name>Class</attribute-class-name>
      </dataset>
      <recombination-prob>0.8</recombination-prob>
      <copy-prob>0.01</copy-prob>
      <listener type=''net.sf.jclec.problem.classification.listener.RuleBaseReporter''>
         <report-dir-name>reports/reportFreitas</report-dir-name>
         <global-report-name>summaryFreitas</global-report-name>
         <report-frequency>10</report-frequency>
      </listener>
   </process>
</experiment>
```

Figure 1: Sample configuration file

Apache commons collections 3.2, Apache commons configuration 1.5, Apache commons lang 2.4, and JUnit 3.8 (for running tests).

Acknowledgments

This research was supported by the Spanish Ministry of Science and Technology project TIN-2011-22408, the Ministry of Education FPU grants AP2010-0041 and AP2010-0042, and FEDER funds.

References

C. C. Bojarczuk, H. S. Lopes, A. A. Freitas, and E. L. Michalkiewicz. A constrained-syntax genetic programming system for discovering classification rules: application to medical data sets. *Artificial Intelligence in Medicine*, 30(1):27–48, 2004.

I. De Falco, A. Della Cioppa, and E. Tarantino. Discovering interesting classification rules with genetic programming. *Applied Soft Computing*, 1(4):257–269, 2001.

P. G. Espejo, S. Ventura, and F. Herrera. A survey on the application of genetic programming to classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 40(2):121–144, 2010.

R. McKay, N. Hoai, P. Whigham, Y. Shan, and M. O'Neill. Grammar-based genetic programming: a survey. *Genetic Programming and Evolvable Machines*, 11:365–396, 2010.

K. C. Tan, A. Tay, T. H. Lee, and C. M. Heng. Mining multiple comprehensible classification rules using genetic programming. In *Proceedings of the Evolutionary Computation on 2002. CEC '02*, volume 2, pages 1302–1307, 2002.

S. Ventura, C. Romero, A. Zafra, J.A. Delgado, and C. Hervás. JCLEC: a Java framework for evolutionary computation. *Soft Computing*, 12:381–392, 2007.

Journal of Machine Learning Research 16 (2015) 495-545 Submitted 9/12; Revised 1/14; Published 3/15

AD^3: Alternating Directions Dual Decomposition for MAP Inference in Graphical Models*

André F. T. Martins ATM@PRIBERAM.PT
Priberam Labs,
Alameda D. Afonso Henriques 41, 2.°, 1000–123 Lisboa, Portugal
and
Instituto de Telecomunicações,
Av. Rovisco Pais 1, 1049–001 Lisboa, Portugal

Mário A. T. Figueiredo MTF@LX.IT.PT
Instituto de Telecomunicações, and Instituto Superior Técnico, Universidade de Lisboa
Av. Rovisco Pais 1, 1049–001 Lisboa, Portugal

Pedro M. Q. Aguiar AGUIAR@ISR.IST.UTL.PT
Instituto de Sistemas e Robótica, and Instituto Superior Técnico, Universidade de Lisboa
Av. Rovisco Pais 1, 1049–001 Lisboa, Portugal

Noah A. Smith NASMITH@CS.CMU.EDU
Eric P. Xing EPXING@CS.CMU.EDU
School of Computer Science, Carnegie Mellon University,
5000 Forbes Ave, Pittsburgh PA 15213–3891, USA

Editor: Tommi Jaakkola

Abstract

We present AD^3, a new algorithm for approximate *maximum a posteriori* (MAP) inference on factor graphs, based on the alternating directions method of multipliers. Like other dual decomposition algorithms, AD^3 has a modular architecture, where local subproblems are solved independently, and their solutions are gathered to compute a global update. The key characteristic of AD^3 is that each local subproblem has a quadratic regularizer, leading to faster convergence, both theoretically and in practice. We provide closed-form solutions for these AD^3 subproblems for binary pairwise factors and factors imposing first-order logic constraints. For arbitrary factors (large or combinatorial), we introduce an active set method which requires only an oracle for computing a local MAP configuration, making AD^3 applicable to a wide range of problems. Experiments on synthetic and real-world problems show that AD^3 compares favorably with the state-of-the-art.

Keywords: MAP inference, graphical models, dual decomposition, alternating directions method of multipliers.

1. Introduction

Graphical models enable compact representations of probability distributions, being widely used in natural language processing (NLP), computer vision, signal processing, and computational biology (Pearl, 1988; Lauritzen, 1996; Koller and Friedman, 2009). When using

*. An earlier version of this work appeared in Martins et al. (2011a).

these models, a central problem is that of inferring the most probable (a.k.a. *maximum a posteriori* – MAP) configuration. Unfortunately, exact MAP inference is an intractable problem for many graphical models of interest in applications, such as those involving non-local features and/or structural constraints. This fact has motivated a significant research effort on approximate techniques.

A class of methods that proved effective for approximate inference is based on linear programming relaxations of the MAP problem (LP-MAP; Schlesinger 1976). Several message-passing and dual decomposition algorithms have been proposed to address the resulting LP problems, taking advantage of the underlying graph structure (Wainwright et al., 2005; Kolmogorov, 2006; Werner, 2007; Komodakis et al., 2007; Globerson and Jaakkola, 2008; Jojic et al., 2010). All these algorithms have a similar consensus-based architecture: they repeatedly perform certain "local" operations in the graph (as outlined in Table 1), until some form of local agreement is achieved. The simplest example is the *projected subgradient dual decomposition* (PSDD) algorithm of Komodakis et al. (2007), which has recently enjoyed great success in NLP applications (see Rush and Collins 2012 and references therein). The major drawback of PSDD is that it is too slow to achieve consensus in large problems, requiring $O(1/\epsilon^2)$ iterations for an ϵ-accurate solution. While block coordinate descent schemes are usually faster to make progress (Globerson and Jaakkola, 2008), they may get stuck in suboptimal solutions, due to the non-smoothness of the dual objective function. Smoothing-based approaches (Jojic et al., 2010; Hazan and Shashua, 2010) do not have these drawbacks, but in turn they typically involve adjusting a "temperature" parameter for trading off the desired precision level and the speed of convergence, and may suffer from numerical instabilities in the near-zero temperature regime.

In this paper, we present a new LP-MAP algorithm called AD^3 (*alternating directions dual decomposition*), which allies the modularity of dual decomposition with the effectiveness of augmented Lagrangian optimization, via the *alternating directions method of multipliers* (Glowinski and Marroco, 1975; Gabay and Mercier, 1976). AD^3 has an iteration bound of $O(1/\epsilon)$, an order of magnitude better than the PSDD algorithm. Like PSDD, AD^3 alternates between a *broadcast* operation, where subproblems are assigned to local workers, and a *gather* operation, where the local solutions are assembled by a controller, which produces an estimate of the global solution. The key difference is that AD^3 regularizes their local subproblems toward these global estimate, which has the effect of speeding up consensus. In many cases of interest, there are closed-form solutions or efficient procedures for solving the AD^3 local subproblems (which are quadratic). For factors lacking such a solution, we introduce an *active set method* which requires only a local MAP decoder (the same requirement as in PSDD). This paves the way for using AD^3 with dense or structured factors.

Our main contributions are:

- We derive AD^3 and establish its convergence properties, blending classical and newer results about ADMM (Eckstein and Bertsekas, 1992; Boyd et al., 2011; Wang and Banerjee, 2012). We show that the algorithm has the same form as the PSDD method of Komodakis et al. (2007), with the local MAP subproblems replaced by quadratic programs. We also show that AD^3 can be wrapped into a branch-and-bound procedure to retrieve the *exact* MAP.

Algorithm	Local Operation
TRW-S (Wainwright et al., 2005; Kolmogorov, 2006)	max-marginals
MPLP (Globerson and Jaakkola, 2008)	max-marginals
PSDD (Komodakis et al., 2007)	MAP
Norm-Product BP (Hazan and Shashua, 2010)	marginals
Accelerated DD (Jojic et al., 2010)	marginals
AD^3 (Martins et al., 2011a)	QP/MAP

Table 1: Several LP-MAP inference algorithms and the kind of the local operations they need to perform at the factors to pass messages and compute beliefs. Some of these operations are the same as the classic loopy BP algorithm, which needs marginals (sum-product variant) or max-marginals (max-product variant). In Section 6, we will see that the quadratic problems (QP) required by AD^3 can be solved as a sequence of local MAP problems.

- We show that these AD^3 subproblems can be solved exactly and efficiently in many cases of interest, including Ising models and a wide range of hard factors representing arbitrary constraints in first-order logic. Up to a logarithmic term, the asymptotic cost in these cases is the same as that of passing messages or doing local MAP inference.

- For factors lacking a closed-form solution of the AD^3 subproblems, we introduce a new *active set method*. All is required is a black box that returns local MAP configurations for each factor (the same requirement of the PSDD algorithm). This paves the way for using AD^3 with large dense or structured factors, based on off-the-shelf combinatorial algorithms (e.g., Viterbi or Chu-Liu-Edmonds).

AD^3 was originally introduced by Martins et al. (2010, 2011a) (then called DD-ADMM). In addition to a considerably more detailed presentation, this paper contains contributions that substantially extend that preliminary work in several directions: the $O(1/\epsilon)$ rate of convergence, the active set method for general factors, and the branch-and-bound procedure for exact MAP inference. It also reports a wider set of experiments and the release of open-source code (available at `http://www.ark.cs.cmu.edu/AD3`), which may be useful to other researchers in the field.

This paper is organized as follows. We start by providing background material: MAP inference in graphical models and its LP-MAP relaxation (Section 2); the PSDD algorithm of Komodakis et al. (2007) (Section 3). In Section 4, we derive AD^3 and analyze its convergence. The AD^3 local subproblems are addressed in Section 5, where closed-form solutions are derived for Ising models and several structural constraint factors. In Section 6, we introduce an active set method to solve the AD^3 subproblems for arbitrary factors. Experiments with synthetic models, as well as in protein design and dependency parsing (Section 7) testify for the success of our approach. Finally, a discussion of related work in presented in Section 8, and Section 9 concludes the paper.

2. Background

We start by providing some background on inference in graphical models.

2.1 Factor Graphs

Let $Y_1, \dots, Y_M$ be random variables describing a structured output, with each Y_i taking values in a finite set $\mathcal{Y}_i$. We follow the common assumption in structured prediction that some of these variables have strong statistical dependencies. In this article, we use *factor graphs* (Tanner, 1981; Kschischang et al., 2001), a convenient way of representing such dependencies that captures directly the factorization assumptions in a model.

Definition 1 (Factor graph) *A factor graph is a bipartite graph $G := (V, F, E)$, comprised of:*

- *a set of* variable nodes $V := \{1, \dots, M\}$*, corresponding to the variables $Y_1, \dots, Y_M$;*
- *a set of* factor nodes F *(disjoint from V);*
- *a set of* edges $E \subseteq V \times F$ *linking variable nodes to factor nodes.*

For notational convenience, we use Latin letters $(i, j, ...)$ and Greek letters $(\alpha, \beta, ...)$ to refer to variable and factor nodes, respectively. We denote by $\partial(\cdot)$ the *neighborhood set* of its node argument, whose cardinality is called the *degree* of the node. Formally, $\partial(i) := \{\alpha \in F \mid (i, \alpha) \in E\}$, for variable nodes, and $\partial(\alpha) := \{i \in V \mid (i, \alpha) \in E\}$ for factor nodes. We use the short notation $\boldsymbol{Y}_\alpha$ to refer to tuples of random variables, which take values on the product set $\mathcal{Y}_\alpha := \prod_{i \in \partial(\alpha)} \mathcal{Y}_i$.

We say that the joint probability distribution of $Y_1, \dots, Y_M$ factors according to the factor graph $G = (V, F, E)$ if it can be written as

$$\mathbb{P}(Y_1 = y_1, \dots, Y_M = y_M) \;\propto\; \exp\left(\sum_{i \in V} \boldsymbol{\theta}_i(y_i) + \sum_{\alpha \in F} \boldsymbol{\theta}_\alpha(\boldsymbol{y}_\alpha)\right), \tag{1}$$

where $\boldsymbol{\theta}_i(\cdot)$ and $\boldsymbol{\theta}_\alpha(\cdot)$ are called, respectively, the *unary* and *higher-order* log-potential functions.[1] To accommodate hard constraints, we allow these functions to take values in $\bar{\mathbb{R}} := \mathbb{R} \cup \{-\infty\}$, but we require them to be *proper* (i.e., they cannot take the value $-\infty$ in their whole domain). Figure 1 shows examples of factor graphs with hard constraint factors (to be studied in detail in Section 5.2).

2.2 MAP Inference

Given a probability distribution specified as in (1), we are interested in finding an assignment with maximal probability (the so-called *MAP assignment/configuration*):

$$\widehat{\boldsymbol{y}}_1, \dots, \widehat{\boldsymbol{y}}_M \;\in\; \arg\max_{y_1, \dots, y_M} \sum_{i \in V} \boldsymbol{\theta}_i(y_i) + \sum_{\alpha \in F} \boldsymbol{\theta}_\alpha(\boldsymbol{y}_\alpha). \tag{2}$$

1. Some authors omit the unary log-potentials, which do not increase generality since they can be absorbed into the higher-order ones. We explicitly state them here since they are frequently used in practice, and their presence highlights a certain symmetry between potentials and marginal variables that will appear in the sequel.

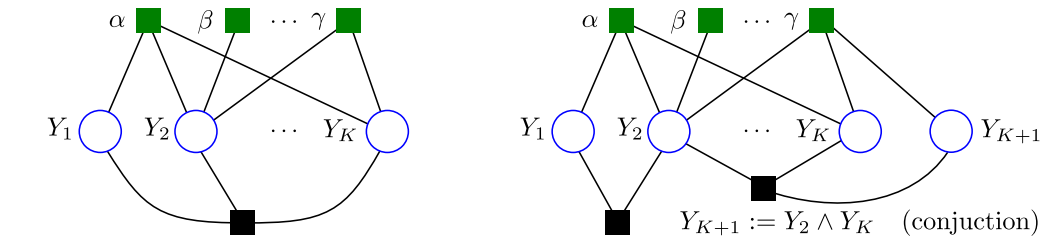


Figure 1: Constrained factor graphs, with soft factors shown as *green* squares above the variable nodes (circles) and hard constraint factors as *black* squares below the variable nodes. Left: a global factor that constrains the set of admissible outputs to a given codebook. Right: examples of declarative constraints; one of them is a factor connecting existing variables to an extra variable, allows scores depending on a logical functions of the former.

In fact, this problem is not specific to probabilistic models: other models, e.g., trained to maximize margin, also lead to maximizations of the form above. Unfortunately, for a general factor graph G, this combinatorial problem is NP-hard (Koller and Friedman, 2009), so one must resort to approximations. In this paper, we address a class of approximations based on linear programming relaxations, described formally in the next section.

Throughout the paper, we will make the following assumption:

Assumption 2 *The MAP problem* (2) *is feasible, i.e., there is at least one assignment* $y_1, \ldots, y_M$ *such that* $\sum_{i \in V} \boldsymbol{\theta}_i(y_i) + \sum_{\alpha \in F} \boldsymbol{\theta}_\alpha(\boldsymbol{y}_\alpha) > -\infty$.

Note that Assumption 2 is substantially weaker than other assumptions made in the literature on graphical models, which sometimes require the solution of to be unique, or the log-potentials to be all finite. We will see in Section 4 that this is all we need for AD^3 to be globally convergent.

2.3 LP-MAP Inference

Schlesinger's linear relaxation (Schlesinger, 1976; Werner, 2007) is the building block for many popular approximate MAP inference algorithms. Let us start by representing the log-potential functions in vector notation, $\boldsymbol{\theta}_i := (\boldsymbol{\theta}_i(y_i))_{y_i \in \mathcal{Y}_i} \in \bar{\mathbb{R}}^{|\mathcal{Y}_i|}$ and $\boldsymbol{\theta}_\alpha := (\boldsymbol{\theta}_\alpha(\boldsymbol{y}_\alpha))_{\boldsymbol{y}_\alpha \in \mathcal{Y}_\alpha} \in \bar{\mathbb{R}}^{|\mathcal{Y}_\alpha|}$. We introduce "local" probability distributions over the variables and factors, represented as vectors of the same size:

$$\boldsymbol{p}_i \in \Delta^{|\mathcal{Y}_i|}, \ \forall i \in V \quad \text{and} \quad \boldsymbol{q}_\alpha \in \Delta^{|\mathcal{Y}_\alpha|}, \ \forall \alpha \in F,$$

where $\Delta^K := \{\boldsymbol{u} \in \mathbb{R}^K \mid \boldsymbol{u} \geq \boldsymbol{0}, \ \boldsymbol{1}^\top \boldsymbol{u} = 1\}$ denotes the K-dimensional probability simplex. We stack these distributions into vectors $\boldsymbol{p}$ and $\boldsymbol{q}$, with dimensions $P := \sum_{i \in V} |\mathcal{Y}_i|$ and $Q := \sum_{\alpha \in F} |\mathcal{Y}_\alpha|$, respectively. If these local probability distributions are "valid" marginal probabilities (i.e., marginals realizable by some global probability distribution

$\mathbb{P}(Y_1, \ldots, Y_M)$), then a necessary (but not sufficient) condition is that they are *locally consistent*. In other words, they must satisfy the following *calibration equations*:

$$\sum_{\boldsymbol{y}_\alpha \sim y_i} q_\alpha(\boldsymbol{y}_\alpha) = p_i(y_i), \qquad \forall y_i \in \mathcal{Y}_i,\ \forall (i,\alpha) \in E, \tag{3}$$

where the notation $\sim$ means that the summation is over all configurations $\boldsymbol{y}_\alpha$ whose ith element equals y_i. Equation (3) can be written in vector notation as $\mathbf{M}_{i\alpha}\boldsymbol{q}_\alpha = \boldsymbol{p}_i,\ \forall (i,\alpha) \in E$, where we define *consistency matrices*

$$\mathbf{M}_{i\alpha}(y_i, \boldsymbol{y}_\alpha) = \begin{cases} 1, & \text{if } \boldsymbol{y}_\alpha \sim y_i \\ 0, & \text{otherwise.} \end{cases}$$

The set of locally consistent distributions forms the *local polytope*:

$$\mathcal{L}(G) = \left\{ (\boldsymbol{p}, \boldsymbol{q}) \in \mathbb{R}^{P+Q} \;\middle|\; \begin{array}{ll} \boldsymbol{q}_\alpha \in \Delta^{|\mathcal{Y}_\alpha|}, & \forall \alpha \in F \\ \mathbf{M}_{i\alpha}\boldsymbol{q}_\alpha = \boldsymbol{p}_i, & \forall (i,\alpha) \in E \end{array} \right\}. \tag{4}$$

We consider the following linear program (the *LP-MAP inference problem*):

$$\boxed{\begin{array}{lll} \textbf{LP-MAP:} & \text{maximize} & \displaystyle\sum_{\alpha \in F} \boldsymbol{\theta}_\alpha{}^\top \boldsymbol{q}_\alpha + \sum_{i \in V} \boldsymbol{\theta}_i{}^\top \boldsymbol{p}_i \\ & \text{with respect to} & (\boldsymbol{p}, \boldsymbol{q}) \in \mathcal{L}(G). \end{array}} \tag{5}$$

If the solution $(\boldsymbol{p}^*, \boldsymbol{q}^*)$ of problem (5) happens to be integral, then each $\boldsymbol{p}_i^*$ and $\boldsymbol{q}_\alpha^*$ will be at corners of the simplex, i.e., they will be indicator vectors of local configurations y_i^* and $\boldsymbol{y}_\alpha^*$, in which case the output $(y_i^*)_{i \in V}$ is guaranteed to be a solution of the MAP decoding problem (2). Under certain conditions—for example, when the factor graph G does not have cycles—problem (5) is guaranteed to have integral solutions. In general, however, the LP-MAP decoding problem (5) is a relaxation of (2). Geometrically, $\mathcal{L}(G)$ is an outer approximation of the *marginal polytope*, defined as the set of valid marginals (Wainwright and Jordan, 2008). This is illustrated in Figure 2.

2.4 LP-MAP Inference Algorithms

While any off-the-shelf LP solver can be used for solving problem (5), specialized algorithms have been designed to exploit the graph structure, achieving superior performance on several benchmarks (Yanover et al., 2006). Some of these algorithms are listed in Table 1. Most of these specialized algorithms belong to two classes: block (dual) coordinate descent, which take the form of *message-passing* algorithms, and projected subgradient algorithms, based on *dual decomposition*.

Block coordinate descent methods address the dual of (5) by alternately optimizing over blocks of coordinates. Examples are max-sum diffusion (Kovalevsky and Koval, 1975; Werner, 2007); max-product sequential tree-reweighted belief propagation (TRW-S, Wainwright et al. 2005; Kolmogorov 2006); and the max-product linear programming algorithm (MPLP; Globerson and Jaakkola 2008). These algorithms work by passing local messages (that require computing *max-marginals*) between factors and variables. Under certain conditions (more stringent than Assumption 2), one may obtain optimality certificates when

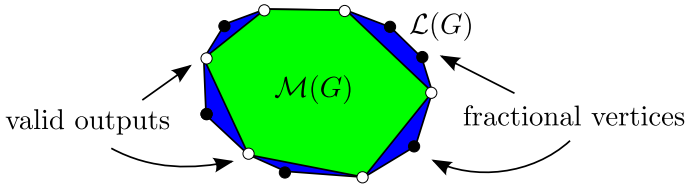


Figure 2: Marginal polytope (in green) and its outer approximation, the local polytope (in blue). Each element of the marginal polytope corresponds to a joint distribution of $Y_1, \ldots, Y_M$, and each vertex corresponds to a configuration $\boldsymbol{y} \in \mathcal{Y}$, having coordinates in $\{0, 1\}$. The local polytope may have additional fractional vertices, with coordinates in $[0, 1]$.

the relaxation is tight. A disadvantage of coordinate descent algorithms is that they may get stuck at suboptimal solutions (Bertsekas et al. 1999, Section 6.3.4), since the dual objective is non-smooth (cf. equation (8) below). An alternative is to optimize the dual with the projected subgradient method, which is globally convergent (Komodakis et al., 2007), and requires computing *local MAP configurations* as its subproblems. Finally, smoothing-based approaches, such as the accelerated dual decomposition method of Jojic et al. (2010) and the norm-product algorithm of Hazan and Shashua (2010), smooth the dual objective with an en tropic regularization term, leading to subproblems that involve computing *local marginals*.

In Section 8, we discuss advantages and disadvantages of these and other LP-MAP inference methods with respect to AD^3.

3. Dual Decomposition with the Projected Subgradient Algorithm

We now describe the *projected subgradient dual decomposition* (PSDD) algorithm proposed by Komodakis et al. (2007). As we will see in Section 4, there is a strong affinity between PSDD and the main focus of this paper, AD^3.

Let us first reparameterize (5) to express it as a consensus problem. For each edge $(i, \alpha) \in E$, we define a potential function $\boldsymbol{\theta}_{i\alpha} := (\boldsymbol{\theta}_{i\alpha}(y_i))_{y_i \in \mathcal{Y}_i}$ that satisfies $\sum_{\alpha \in \partial(i)} \boldsymbol{\theta}_{i\alpha} = \boldsymbol{\theta}_i$; a trivial choice is $\boldsymbol{\theta}_{i\alpha} = |\partial(i)|^{-1}\boldsymbol{\theta}_i$, which spreads the unary potentials evenly across the factors. Since we have a equality constraint $\boldsymbol{p}_i = \mathbf{M}_{i\alpha}\boldsymbol{q}_\alpha$, problem (5) is equivalent to the following *primal formulation*:

$$\boxed{\begin{aligned} \textbf{LP-MAP-P:} \quad \text{maximize} \quad & \sum_{\alpha \in F} \left(\boldsymbol{\theta}_\alpha + \sum_{i \in \partial(\alpha)} \mathbf{M}_{i\alpha}^\top \boldsymbol{\theta}_{i\alpha} \right)^\top \boldsymbol{q}_\alpha \\ \text{with respect to} \quad & \boldsymbol{p} \in \mathbb{R}^P, \quad \boldsymbol{q}_\alpha \in \Delta^{|\mathcal{Y}_\alpha|}, \forall \alpha \in F, \\ \text{subject to} \quad & \mathbf{M}_{i\alpha}\boldsymbol{q}_\alpha = \boldsymbol{p}_i, \ \forall (i, \alpha) \in E. \end{aligned}} \tag{6}$$

Note that, although the $\boldsymbol{p}$-variables do not appear in the objective of (6), they play a fundamental role through the constraints in the last line, which are necessary to ensure

that the marginals encoded in the $\boldsymbol{q}$-variables are consistent on their overlaps. Indeed, it is this set of constraints that complicate the optimization problem, which would otherwise be separable into independent subproblems, one per factor. Introducing Lagrange multipliers $\boldsymbol{\lambda}_{i\alpha} := (\boldsymbol{\lambda}_{i\alpha}(y_i))_{y_i \in \mathcal{Y}_i}$ for each of these equality constraints leads to the Lagrangian function

$$L(\boldsymbol{q}, \boldsymbol{p}, \boldsymbol{\lambda}) = \sum_{\alpha \in F} \left(\boldsymbol{\theta}_\alpha + \sum_{i \in \partial(\alpha)} \mathbf{M}_{i\alpha}^\top (\boldsymbol{\theta}_{i\alpha} + \boldsymbol{\lambda}_{i\alpha}) \right)^\top \boldsymbol{q}_\alpha - \sum_{(i,\alpha) \in E} \boldsymbol{\lambda}_{i\alpha}{}^\top \boldsymbol{p}_i, \tag{7}$$

the maximization of which w.r.t. $\boldsymbol{q}$ and $\boldsymbol{p}$ will yield the (Lagrangian) dual objective. Since the $\boldsymbol{p}$-variables are unconstrained, we have

$$\max_{\boldsymbol{q}, \boldsymbol{p}} L(\boldsymbol{q}, \boldsymbol{p}, \boldsymbol{\lambda}) = \begin{cases} g(\boldsymbol{\lambda}) & \text{if } \boldsymbol{\lambda} \in \Lambda, \\ +\infty & \text{otherwise,} \end{cases}$$

and we arrive at the following *dual formulation*:

$$\boxed{\begin{array}{lll} \textbf{LP-MAP-D:} & \text{minimize} & g(\boldsymbol{\lambda}) := \sum_{\alpha \in F} g_\alpha(\boldsymbol{\lambda}) \\ & \text{with respect to} & \boldsymbol{\lambda} \in \Lambda, \end{array}} \tag{8}$$

where $\Lambda := \left\{ \boldsymbol{\lambda} \mid \sum_{\alpha \in \partial(i)} \boldsymbol{\lambda}_{i\alpha} = \mathbf{0},\ \forall i \in V \right\}$ is a linear subspace, and each $g_\alpha(\boldsymbol{\lambda})$ is the solution of a *local subproblem*:

$$\begin{aligned} g_\alpha(\boldsymbol{\lambda}) &:= \max_{\boldsymbol{q}_\alpha \in \Delta^{|\mathcal{Y}_\alpha|}} \left(\boldsymbol{\theta}_\alpha + \sum_{i \in \partial(\alpha)} \mathbf{M}_{i\alpha}^\top (\boldsymbol{\theta}_{i\alpha} + \boldsymbol{\lambda}_{i\alpha}) \right)^\top \boldsymbol{q}_\alpha \\ &= \max_{\boldsymbol{y}_\alpha \in \mathcal{Y}_\alpha} \left(\boldsymbol{\theta}_\alpha(\boldsymbol{y}_\alpha) + \sum_{i \in \partial(\alpha)} (\boldsymbol{\theta}_{i\alpha}(y_i) + \boldsymbol{\lambda}_{i\alpha}(y_i)) \right); \end{aligned} \tag{9}$$

the last equality is justified by the fact that maximizing a linear objective over the probability simplex gives the largest component of the score vector. Note that the local subproblem (9) can be solved by a ComputeMAP procedure, which receives unary potentials $\boldsymbol{\xi}_{i\alpha}(y_i) := \boldsymbol{\theta}_{i\alpha}(y_i) + \boldsymbol{\lambda}_{i\alpha}(y_i)$ and factor potentials $\boldsymbol{\theta}_\alpha(\boldsymbol{y}_\alpha)$ (eventually structured) and returns the MAP $\widehat{\boldsymbol{y}}_\alpha$.

Problem (8) is often referred to as the *master* or *controller*, and each local subproblem (9) as a *slave* or *worker*. The master problem (8) can be solved with a *projected subgradient algorithm*.[2] By Danskin's rule (Bertsekas et al., 1999, p. 717), a subgradient of g_α is readily given by

$$\frac{\partial g_\alpha(\boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}_{i\alpha}} = \mathbf{M}_{i\alpha} \widehat{\boldsymbol{q}}_\alpha, \quad \forall (i, \alpha) \in E;$$

and the projection onto Λ amounts to a centering operation. Putting these pieces together yields Algorithm 1. At each iteration, the algorithm broadcasts the current Lagrange multipliers to all the factors. Each factor adjusts its internal unary log-potentials (line 6) and

2. A slightly different formulation is presented by Sontag et al. (2011) which yields a subgradient algorithm with no projection.

Algorithm 1 PSDD Algorithm (Komodakis et al., 2007)

1: **input:** graph G, parameters $\boldsymbol{\theta}$, maximum number of iterations T, step sizes $(\eta_t)_{t=1}^T$
2: for each $(i,\alpha) \in E$, choose $\boldsymbol{\theta}_{i\alpha}$ such that $\sum_{\alpha\in\partial(i)} \boldsymbol{\theta}_{i\alpha} = \boldsymbol{\theta}_i$
3: initialize $\boldsymbol{\lambda} = \mathbf{0}$
4: **for** $t = 1$ **to** T **do**
5: **for each** factor $\alpha \in F$ **do**
6: set unary log-potentials $\boldsymbol{\xi}_{i\alpha} := \boldsymbol{\theta}_{i\alpha} + \boldsymbol{\lambda}_{i\alpha}$, for $i \in \partial(\alpha)$
7: set $\widehat{\boldsymbol{q}}_\alpha := \textsc{ComputeMap}(\boldsymbol{\theta}_\alpha + \sum_{i\in\partial(\alpha)} \mathbf{M}_{i\alpha}^\top \boldsymbol{\xi}_{i\alpha})$
8: set $\widehat{\boldsymbol{q}}_{i\alpha} := \mathbf{M}_{i\alpha}\widehat{\boldsymbol{q}}_\alpha$, for $i \in \partial(\alpha)$
9: **end for**
10: compute average $\boldsymbol{p}_i := |\partial(i)|^{-1} \sum_{\alpha\in\partial(i)} \widehat{\boldsymbol{q}}_{i\alpha}$ for each $i \in V$
11: update $\boldsymbol{\lambda}_{i\alpha} := \boldsymbol{\lambda}_{i\alpha} - \eta_t\,(\widehat{\boldsymbol{q}}_{i\alpha} - \boldsymbol{p}_i)$ for each $(i,\alpha) \in E$
12: **end for**
13: **output:** dual variable $\boldsymbol{\lambda}$ and upper bound $g(\boldsymbol{\lambda})$

invokes the ComputeMap procedure (line 7).[3] The solutions achieved by each factor are then gathered and averaged (line 10), and the Lagrange multipliers are updated with step size η_t (line 11). The two following propositions establish the convergence properties of Algorithm 1.

Proposition 3 (Convergence rate) *If the non-negative step size sequence $(\eta_t)_{t\in\mathbb{N}}$ is diminishing and nonsummable ($\lim \eta_t = 0$ and $\sum_{t=1}^\infty \eta_t = \infty$), then Algorithm 1 converges to the solution $\boldsymbol{\lambda}^*$ of LP-MAP-D (8). Furthermore, after $T = O(1/\epsilon^2)$ iterations, we have $g(\boldsymbol{\lambda}^{(T)}) - g(\boldsymbol{\lambda}^*) \leq \epsilon$.*

Proof: This is a property of projected subgradient algorithms (see, e.g., Bertsekas et al. 1999). ■

Proposition 4 (Certificate of optimality) *If, at some iteration of Algorithm 1, all the local subproblems are in agreement (i.e., if $\widehat{\boldsymbol{q}}_{i\alpha} = \boldsymbol{p}_i$ after line 10, for all $i \in V$), then: (i) $\boldsymbol{\lambda}$ is a solution of LP-MAP-D (8); (ii) $\boldsymbol{p}$ is binary-valued and a solution of both LP-MAP-P and MAP.*

Proof: If all local subproblems are in agreement, then a vacuous update will occur in line 11, and no further changes will occur. Since the algorithm is guaranteed to converge, the current $\boldsymbol{\lambda}$ is optimal. Also, if all local subproblems are in agreement, the averaging in line 10 necessarily yields a binary vector $\boldsymbol{p}$. Since any binary solution of LP-MAP is also a solution of MAP, the result follows. ■

Propositions 3–4 imply that, if the LP-MAP relaxation is tight, then Algorithm 1 will eventually yield the exact MAP configuration along with a certificate of optimality. According to Proposition 3, even if the relaxation is not tight, Algorithm 1 still converges to

3. Note that, if the factor log-potentials $\boldsymbol{\theta}_\alpha$ have special structure (e.g., if the factor is itself combinatorial, such as a sequence or a tree model), then this structure is preserved since only the internal unary log-potentials are changed. Therefore, if evaluating $\textsc{ComputeMap}(\boldsymbol{\theta}_\alpha)$ is tractable, so is evaluating $\textsc{ComputeMap}(\boldsymbol{\theta}_\alpha + \sum_{i\in\partial(\alpha)} \mathbf{M}_{i\alpha}^\top \boldsymbol{\xi}_{i\alpha})$.

a solution of LP-MAP. Unfortunately, in large graphs with many overlapping factors, it has been observed that convergence can be quite slow in practice (Martins et al., 2011b). This is not surprising, given that it attempts to reach a consensus among all overlapping components; the larger this number, the harder it is to achieve consensus. We describe in the next section another LP-MAP decoder (AD^3) with a faster convergence rate.

4. Alternating Directions Dual Decomposition (AD^3)

AD^3 avoids some of the weaknesses of PSDD by replacing the subgradient method with the *alternating directions method of multipliers* (ADMM). Before going into a formal derivation, let us go back to the PSDD algorithm to pinpoint the crux of its weaknesses. It resides in two aspects:

1. The dual objective function $g(\boldsymbol{\lambda})$ is *non-smooth*, this being why "subgradients" are used instead of "gradients." It is well-known that non-smooth optimization lacks some of the good properties of its smooth counterpart. Namely, there is no guarantee of monotonic improvement in the objective (see Bertsekas et al. 1999, p. 611). Ensuring convergence requires using a diminishing step size sequence, which leads to slow convergence rates. In fact, as stated in Proposition 3, $O(1/\epsilon^2)$ iterations are required to guarantee ϵ-accuracy.

2. A close look at Algorithm 1 reveals that the consensus is promoted solely by the Lagrange multipliers (line 6). These can be regarded as "price adjustments" that are made at each iteration and lead to a reallocation of resources. However, no "memory" exists about past allocations or adjustments, so the workers never know how far they are from consensus. One may suspect that a smarter use of these quantities may accelerate convergence.

The first of these aspects has been addressed by the accelerated dual decomposition method of Jojic et al. (2010), which improves the iteration bound to $O(1/\epsilon)$; we discuss that work further in Section 8. We will see that AD^3 also yields a $O(1/\epsilon)$ iteration bound with some additional advantages. The second aspect is addressed by AD^3 by broadcasting *the current global solution* in addition to the Lagrange multipliers, allowing the workers to regularize their subproblems toward that solution.

4.1 Augmented Lagrangians and the Alternating Directions Method of Multipliers

Let us start with a brief overview of augmented Lagrangian methods. Consider the following general convex optimization problem with equality constraints:

$$\begin{array}{ll} \text{maximize} & f_1(\boldsymbol{q}) + f_2(\boldsymbol{p}) \\ \text{with respect to} & \boldsymbol{q} \in \mathcal{Q}, \boldsymbol{p} \in \mathcal{P} \\ \text{subject to} & \mathbf{A}\boldsymbol{q} + \mathbf{B}\boldsymbol{p} = \boldsymbol{c}, \end{array} \tag{10}$$

where $\mathcal{Q} \subseteq \mathbb{R}^P$ and $\mathcal{P} \subseteq \mathbb{R}^Q$ are convex sets and $f_1 : \mathcal{Q} \to \bar{\mathbb{R}}$ and $f_2 : \mathcal{P} \to \bar{\mathbb{R}}$ are concave functions. Note that the LP-MAP problem stated in (6) has this form. For any $\eta \geq 0$,

consider the problem

$$\begin{array}{ll} \text{maximize} & f_1(\boldsymbol{q}) + f_2(\boldsymbol{p}) - \frac{\eta}{2}\|\mathbf{A}\boldsymbol{q} + \mathbf{B}\boldsymbol{p} - \boldsymbol{c}\|^2 \\ \text{with respect to} & \boldsymbol{q} \in \mathcal{Q}, \boldsymbol{p} \in \mathcal{P} \\ \text{subject to} & \mathbf{A}\boldsymbol{q} + \mathbf{B}\boldsymbol{p} = \boldsymbol{c}, \end{array} \tag{11}$$

which differs from (10) in the extra term penalizing violations of the equality constraints; since this term vanishes at feasibility, the two problems have the same solution. The Lagrangian of (11),

$$L_\eta(\boldsymbol{q}, \boldsymbol{p}, \boldsymbol{\lambda}) = f_1(\boldsymbol{q}) + f_2(\boldsymbol{p}) + \boldsymbol{\lambda}^\top(\mathbf{A}\boldsymbol{q} + \mathbf{B}\boldsymbol{p} - \boldsymbol{c}) - \frac{\eta}{2}\|\mathbf{A}\boldsymbol{q} + \mathbf{B}\boldsymbol{p} - \boldsymbol{c}\|^2,$$

is called the *η-augmented Lagrangian* of (10). The so-called *augmented Lagrangian* methods (Bertsekas et al., 1999, Section 4.2) address problem (10) by seeking a saddle point of L_{η_t}, for some sequence $(\eta_t)_{t\in\mathbb{N}}$. The earliest instance is the *method of multipliers* (Hestenes, 1969; Powell, 1969), which alternates between a joint update of $\boldsymbol{q}$ and $\boldsymbol{p}$ through

$$(\boldsymbol{q}^{t+1}, \boldsymbol{p}^{t+1}) \;:=\; \arg\max_{\boldsymbol{q},\boldsymbol{p}} \{L_{\eta_t}(\boldsymbol{q}, \boldsymbol{p}, \boldsymbol{\lambda}^t) \mid \boldsymbol{q} \in \mathcal{Q}, \boldsymbol{p} \in \mathcal{P}\} \tag{12}$$

and a gradient update of the Lagrange multiplier,

$$\boldsymbol{\lambda}^{t+1} \;:=\; \boldsymbol{\lambda}^t - \eta_t(\mathbf{A}\boldsymbol{q}^{t+1} + \mathbf{B}\boldsymbol{p}^{t+1} - \boldsymbol{c}).$$

Under some conditions, this method is convergent, and even superlinear, if the sequence $(\eta_t)_{t\in\mathbb{N}}$ is increasing (Bertsekas et al. 1999, Section 4.2). A shortcoming of this method is that problem (12) may be difficult, since the penalty term of the augmented Lagrangian couples the variables $\boldsymbol{p}$ and $\boldsymbol{q}$. The *alternating directions method of multipliers* (ADMM) avoids this shortcoming, by replacing the joint optimization (12) by a single block Gauss-Seidel-type step:

$$\boldsymbol{q}^{t+1} := \arg\max_{\boldsymbol{q}\in\mathcal{Q}} L_{\eta_t}(\boldsymbol{q}, \boldsymbol{p}^t, \boldsymbol{\lambda}^t) = \arg\max_{\boldsymbol{q}\in\mathcal{Q}} f_1(\boldsymbol{q}) + (\mathbf{A}^\top\boldsymbol{\lambda}^t)^\top\boldsymbol{q} - \frac{\eta_t}{2}\|\mathbf{A}\boldsymbol{q} + \mathbf{B}\boldsymbol{p}^t - \boldsymbol{c}\|^2, \tag{13}$$

$$\boldsymbol{p}^{t+1} := \arg\max_{\boldsymbol{p}\in\mathcal{P}} L_{\eta_t}(\boldsymbol{q}^{t+1}, \boldsymbol{p}, \boldsymbol{\lambda}^t) = \arg\max_{\boldsymbol{p}\in\mathcal{P}} f_2(\boldsymbol{p}) + (\mathbf{B}^\top\boldsymbol{\lambda}^t)^\top\boldsymbol{p} - \frac{\eta_t}{2}\|\mathbf{A}\boldsymbol{q}^{t+1} + \mathbf{B}\boldsymbol{p} - \boldsymbol{c}\|^2. \tag{14}$$

In general, problems (13)–(14) are simpler than the joint maximization in (12). ADMM was proposed by Glowinski and Marroco (1975) and Gabay and Mercier (1976) and is related to other optimization methods, such as Douglas-Rachford splitting (Eckstein and Bertsekas, 1992) and proximal point methods (see Boyd et al. 2011 for an historical overview).

4.2 Derivation of AD^3

Our LP-MAP-P problem (6) can be cast into the form (10) by proceeding as follows:

- let $\mathcal{Q}$ in (10) be the Cartesian product of simplices, $\mathcal{Q} := \prod_{\alpha\in F} \Delta^{|\mathcal{Y}_\alpha|}$, and $\mathcal{P} := \mathbb{R}^P$;
- let $f_1(\boldsymbol{q}) := \sum_{\alpha\in F} \left(\boldsymbol{\theta}_\alpha + \sum_{i\in\partial(\alpha)} \mathbf{M}_{i\alpha}^\top \boldsymbol{\theta}_{i\alpha}\right)^\top \boldsymbol{q}_\alpha$ and $f_2 :\equiv 0$;

- let $\mathbf{A}$ in (10) be a $R \times Q$ block-diagonal matrix, where $R = \sum_{(i,\alpha)\in E} |\mathcal{Y}_i|$, with one block per factor, which is a vertical concatenation of the matrices $\{\mathbf{M}_{i\alpha}\}_{i\in\partial(\alpha)}$;

- let $-\mathbf{B}$ be a $R \times P$ matrix of grid-structured blocks, where the block in the (i,α)th row and the ith column is a negative identity matrix of size $|\mathcal{Y}_i|$, and all the other blocks are zero;

- let $\boldsymbol{c} := 0$.

The η-augmented Lagrangian associated with (6) is

$$L_\eta(\boldsymbol{q},\boldsymbol{p},\boldsymbol{\lambda}) = \sum_{\alpha\in F}\left(\boldsymbol{\theta}_\alpha + \sum_{i\in\partial(\alpha)} \mathbf{M}_{i\alpha}^\top(\boldsymbol{\theta}_{i\alpha}+\boldsymbol{\lambda}_{i\alpha})\right)^\top \boldsymbol{q}_\alpha - \sum_{(i,\alpha)\in E} \boldsymbol{\lambda}_{i\alpha}{}^\top \boldsymbol{p}_i - \frac{\eta}{2}\sum_{(i,\alpha)\in E}\|\mathbf{M}_{i\alpha}\boldsymbol{q}_\alpha - \boldsymbol{p}_i\|^2.$$

This is the standard Lagrangian (7) plus the Euclidean penalty term. The ADMM updates are

$$\textbf{Broadcast:}\quad \boldsymbol{q}^{(t)} := \arg\max_{\boldsymbol{q}\in\mathcal{Q}} L_{\eta_t}(\boldsymbol{q},\boldsymbol{p}^{(t-1)},\boldsymbol{\lambda}^{(t-1)}), \tag{15}$$

$$\textbf{Gather:}\quad \boldsymbol{p}^{(t)} := \arg\max_{\boldsymbol{p}\in\mathbb{R}^P} L_{\eta_t}(\boldsymbol{q}^{(t)},\boldsymbol{p},\boldsymbol{\lambda}^{(t-1)}), \tag{16}$$

$$\textbf{Multiplier update:}\quad \boldsymbol{\lambda}_{i\alpha}^{(t)} := \boldsymbol{\lambda}_{i\alpha}^{(t-1)} - \eta_t\left(\mathbf{M}_{i\alpha}\boldsymbol{q}_\alpha^{(t)} - \boldsymbol{p}_i^{(t)}\right), \forall (i,\alpha)\in E. \tag{17}$$

We next analyze the broadcast and gather steps, and prove a proposition about the multiplier update.

4.2.1 Broadcast Step

The maximization (15) can be carried out in parallel at the factors, as in PSDD. The only difference is that, instead of a local MAP computation, each worker now needs to solve a *quadratic program* of the form:

$$\boxed{\max_{\boldsymbol{q}_\alpha\in\Delta^{|\mathcal{Y}_\alpha|}}\left(\boldsymbol{\theta}_\alpha + \sum_{i\in\partial(\alpha)}\mathbf{M}_{i\alpha}^\top(\boldsymbol{\theta}_{i\alpha}+\boldsymbol{\lambda}_{i\alpha})\right)^\top \boldsymbol{q}_\alpha - \frac{\eta}{2}\sum_{i\in\partial(\alpha)}\|\mathbf{M}_{i\alpha}\boldsymbol{q}_\alpha - \boldsymbol{p}_i\|^2.} \tag{18}$$

This differs from the linear subproblem (9) of PSDD by the inclusion of an Euclidean penalty term, which penalizes deviations from the global consensus. In Sections 5 and 6, we will give procedures to solve these local subproblems.

4.2.2 Gather Step

The solution of problem (16) has a closed form. Indeed, this problem is separable into independent optimizations, one for each $i \in V$; defining $\boldsymbol{q}_{i\alpha} := \mathbf{M}_{i\alpha}\boldsymbol{q}_\alpha$,

$$\begin{aligned}\boldsymbol{p}_i^{(t)} &:= \arg\min_{\boldsymbol{p}_i \in \mathbb{R}^{|\mathcal{Y}_i|}} \sum_{\alpha \in \partial(i)} \left(\boldsymbol{p}_i - \left(\boldsymbol{q}_{i\alpha} - \eta_t^{-1}\boldsymbol{\lambda}_{i\alpha}\right)\right)^2 \\ &= |\partial(i)|^{-1} \sum_{\alpha \in \partial(i)} \left(\boldsymbol{q}_{i\alpha} - \eta_t^{-1}\boldsymbol{\lambda}_{i\alpha}\right) \\ &= \frac{1}{|\partial(i)|} \sum_{\alpha \in \partial(i)} \boldsymbol{q}_{i\alpha}.\end{aligned}$$

The equality in the last line is due to the following proposition:

Proposition 5 *The sequence $\boldsymbol{\lambda}^{(1)}, \boldsymbol{\lambda}^{(2)}, \ldots$ produced by the updates (15)–(17) is dual feasible, i.e., we have $\boldsymbol{\lambda}^{(t)} \in \Lambda$ for every t, with Λ as in (8).*

Proof: We have:

$$\begin{aligned}\sum_{\alpha \in \partial(i)} \boldsymbol{\lambda}_{i\alpha}^{(t)} &= \sum_{\alpha \in \partial(i)} \boldsymbol{\lambda}_{i\alpha}^{(t-1)} - \eta_t \left(\sum_{\alpha \in \partial(i)} \boldsymbol{q}_{i\alpha}^{(t)} - |\partial(i)|\boldsymbol{p}_i^{(t)}\right) \\ &= \sum_{\alpha \in \partial(i)} \boldsymbol{\lambda}_{i\alpha}^{(t-1)} - \eta_t \left(\sum_{\alpha \in \partial(i)} \boldsymbol{q}_{i\alpha}^{(t)} - \sum_{\alpha \in \partial(i)} \left(\boldsymbol{q}_{i\alpha}^{(t)} - \eta_t^{-1}\boldsymbol{\lambda}_{i\alpha}^{(t-1)}\right)\right) = \mathbf{0}.\end{aligned}$$

■

Assembling all these pieces together leads to AD^3 (Algorithm 2), where we use a fixed step size η. Notice that AD^3 retains the modular structure of PSDD (Algorithm 1). The key difference is that AD^3 also broadcasts the current global solution to the workers, allowing them to regularize their subproblems toward that solution, thus speeding up the consensus. This is embodied in the procedure SolveQP (line 7), which replaces ComputeMAP of Algorithm 1.

4.3 Convergence Analysis

Before proving the convergence of AD^3, we start with a basic result.

Proposition 6 (Existence of a Saddle Point) *Under Assumption 2, we have the following properties (regardless of the choice of log-potentials):*

1. *LP-MAP-P (6) is primal-feasible;*
2. *LP-MAP-D (8) is dual-feasible;*
3. *The Lagrangian function $L(\boldsymbol{q}, \boldsymbol{p}, \boldsymbol{\lambda})$ has a saddle point $(\boldsymbol{q}^*, \boldsymbol{p}^*, \boldsymbol{\lambda}^*) \in \mathcal{Q} \times \mathcal{P} \times \Lambda$, where $(\boldsymbol{q}^*, \boldsymbol{p}^*)$ is a solution of LP-MAP-P and $\boldsymbol{\lambda}^*$ is a solution of LP-MAP-D.*

Algorithm 2 Alternating Directions Dual Decomposition (AD^3)

input: graph G, parameters $\boldsymbol{\theta}$, penalty constant η
initialize $\boldsymbol{p}$ uniformly (i.e., $p_i(y_i) = 1/|\mathcal{Y}_i|$, $\forall i \in V, y_i \in \mathcal{Y}_i$)
initialize $\boldsymbol{\lambda} = \mathbf{0}$
repeat
  for each factor $\alpha \in F$ **do**
    set unary log-potentials $\boldsymbol{\xi}_{i\alpha} := \boldsymbol{\theta}_{i\alpha} + \boldsymbol{\lambda}_{i\alpha}$, for $i \in \partial(\alpha)$
    set $\widehat{\boldsymbol{q}}_\alpha := \textsc{SolveQP}\left(\boldsymbol{\theta}_\alpha + \sum_{i\in\partial(\alpha)} \mathbf{M}_{i\alpha}^\top \boldsymbol{\xi}_{i\alpha},\ (\boldsymbol{p}_i)_{i\in\partial(\alpha)}\right)$
    set $\widehat{\boldsymbol{q}}_{i\alpha} := \mathbf{M}_{i\alpha}\widehat{\boldsymbol{q}}_\alpha$, for $i \in \partial(\alpha)$
  end for
  compute average $\boldsymbol{p}_i := |\partial(i)|^{-1} \sum_{\alpha\in\partial(i)} \widehat{\boldsymbol{q}}_{i\alpha}$ for each $i \in V$
  update $\boldsymbol{\lambda}_{i\alpha} := \boldsymbol{\lambda}_{i\alpha} - \eta\,(\widehat{\boldsymbol{q}}_{i\alpha} - \boldsymbol{p}_i)$ for each $(i,\alpha) \in E$
until convergence
output: primal variables $\boldsymbol{p}$ and $\boldsymbol{q}$, dual variable $\boldsymbol{\lambda}$, upper bound $g(\boldsymbol{\lambda})$

Proof: Property 1 follows directly from Assumption 2 and the fact that LP-MAP is a relaxation of MAP. To prove properties 2–3, define first the set of structural constraints $\bar{\mathcal{Q}} := \prod_{\alpha\in F} \bar{\mathcal{Q}}_\alpha$, where $\bar{\mathcal{Q}}_\alpha := \{\boldsymbol{q}_\alpha \in \Delta^{|\mathcal{Y}_\alpha|} \mid \boldsymbol{q}_\alpha(\boldsymbol{y}_\alpha) = 0, \forall \boldsymbol{y}_\alpha \text{ s.t. } \theta_\alpha(\boldsymbol{y}_\alpha) = -\infty\}$ are truncated probability simplices (hence convex). Since all log-potential functions are proper (due to Assumption 2), we have that each $\bar{\mathcal{Q}}_\alpha$ is non-empty, and therefore $\bar{\mathcal{Q}}$ has non-empty relative interior. As a consequence, the refined Slater's condition (Boyd and Vandenberghe, 2004, §5.2.3) holds; let $(q^*, p^*) \in \bar{\mathcal{Q}} \times \mathcal{P}$ be a primal feasible solution of LP-MAP-P, which exists by virtue of property 1. Then, the KKT optimality conditions imply the existence of a $\boldsymbol{\lambda}^*$ such that $(\boldsymbol{q}^*, \boldsymbol{p}^*, \boldsymbol{\lambda}^*)$ is a saddle point of the Lagrangian function L, i.e., $L(\boldsymbol{q}, \boldsymbol{p}, \boldsymbol{\lambda}^*) \leq L(\boldsymbol{q}^*, \boldsymbol{p}^*, \boldsymbol{\lambda}^*) \leq L(\boldsymbol{q}^*, \boldsymbol{p}^*, \boldsymbol{\lambda})$ holds for all $\boldsymbol{q}, \boldsymbol{p}, \boldsymbol{\lambda}$. Naturally, we must have $\boldsymbol{\lambda}^* \in \Lambda$, otherwise $L(.,.,\boldsymbol{\lambda}^*)$ would be unbounded with respect to $\boldsymbol{p}$. ■

We are now ready to show the convergence of AD^3, which follows directly from the general convergence properties of ADMM. Remarkably, unlike in PSDD, convergence is ensured with a fixed step size, therefore no annealing is required.

Proposition 7 (Convergence of AD^3) *Let $(\boldsymbol{q}^{(t)}, \boldsymbol{p}^{(t)}, \boldsymbol{\lambda}^{(t)})_t$ be the sequence of iterates produced by Algorithm 2 with a fixed $\eta_t = \eta$. Then the following holds:*

1. *primal feasibility of LP-MAP-P* (6) *is achieved in the limit, i.e.,*

$$\|\mathbf{M}_{i\alpha}\boldsymbol{q}_\alpha^{(t)} - \boldsymbol{p}_i^{(t)}\| \to \mathbf{0}, \quad \forall (i,\alpha) \in E;$$

2. *the primal objective sequence $\left(\sum_{i\in V} \boldsymbol{\theta}_i{}^\top \boldsymbol{p}_i^{(t)} + \sum_{\alpha\in F} \boldsymbol{\theta}_\alpha{}^\top \boldsymbol{q}_\alpha^{(t)}\right)_{t\in\mathbb{N}}$ converges to the solution of LP-MAP-P* (6)*;*

3. *the dual sequence $(\boldsymbol{\lambda}^{(t)})_{t\in\mathbb{N}}$ converges to a solution of the dual LP-MAP-D* (8)*; moreover, this sequence is dual feasible, i.e., it is contained in Λ. Thus, $g(\boldsymbol{\lambda}^{(t)})$ in* (8) *approaches the optimum from above.*

Proof: See Boyd et al. (2011, Appendix A) for a simple proof of the convergence of ADMM in the form (10), from which 1, 2, and the first part of 3 follow immediately. The two assumptions stated in Boyd et al. (2011, p.16) are met: denoting by $\iota_{\mathcal{Q}}$ the indicator function of the set $\mathcal{Q}$, which evaluates to zero in $\mathcal{Q}$ and to $-\infty$ outside $\mathcal{Q}$, we have that functions $f_1 + \iota_{\mathcal{Q}}$ and f_2 are closed proper convex (since the log-potential functions are proper and f_1 is closed proper convex), and the unaugmented Lagrangian has a saddle point (see property 3 in Proposition 6). Finally, the last part of statement 3 follows from Proposition 5. ■

The next proposition, proved in Appendix A, states the $O(1/\epsilon)$ iteration bound of AD^3, which is better than the $O(1/\epsilon^2)$ bound of PSDD.

Proposition 8 (Convergence rate of AD^3) *Assume the conditions of Proposition 7. Let $\boldsymbol{\lambda}^*$ be a solution of the dual problem (8), $\bar{\boldsymbol{\lambda}}_T := \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{\lambda}^{(t)}$ be the "averaged" Lagrange multipliers after T iterations of AD^3, and $g(\bar{\boldsymbol{\lambda}}_T)$ the corresponding estimate of the dual objective (an upper bound). Then, $g(\bar{\boldsymbol{\lambda}}_T) - g(\boldsymbol{\lambda}^*) \leq \epsilon$ after $T \leq C/\epsilon$ iterations, where C is a constant satisfying*

$$\begin{aligned} C &\leq \frac{5\eta}{2} \sum_{i \in V} |\partial(i)| \times (1 - |\mathcal{Y}_i|^{-1}) + \frac{5}{2\eta} \|\boldsymbol{\lambda}^*\|^2 \\ &\leq \frac{5\eta}{2} |E| + \frac{5}{2\eta} \|\boldsymbol{\lambda}^*\|^2. \end{aligned} \tag{19}$$

As expected, the bound (19) increases with the number of overlapping variables, quantified by the number of edges $|E|$, and the magnitude of the optimal dual vector $\boldsymbol{\lambda}^*$. Note that if there is a good estimate of $\|\boldsymbol{\lambda}^*\|$, then (19) can be used to choose a step size η that minimizes the bound—the optimal step size is $\eta = \|\boldsymbol{\lambda}^*\| \times |E|^{-1/2}$, which would lead to $T \leq 5\epsilon^{-1}|E|^{1/2}\|\boldsymbol{\lambda}^*\|$. In fact, although Proposition 7 guarantees convergence for any choice of η, we observed in practice that this parameter has a strong impact on the behavior of the algorithm. In our experiments, we dynamically adjust η in earlier iterations using the heuristic described in Boyd et al. (2011, Section 3.4.1), and freeze it afterwards, not to compromise convergence.

4.4 Stopping Conditions and Implementation Details

We next establish stopping conditions for AD^3 and discuss some implementation details that can provide significant speed-ups.

4.4.1 Primal and Dual Residuals

Since the AD^3 iterates are dual feasible, it is also possible to check the conditions in Proposition 4 to obtain optimality certificates, as in PSDD. Moreover, even when the LP-MAP relaxation is not tight, AD^3 can provide stopping conditions by keeping track of primal and dual residuals, as described in Boyd et al. (2011, §3.3), based on which it is possible to obtain certificates, not only for the primal solution (if the relaxation is tight), but also to

terminate when a near optimal relaxed primal solution has been found.[4] This is an important advantage over PSDD, which is unable to provide similar stopping conditions, and is usually stopped rather arbitrarily after a given number of iterations.

The *primal residual* $r_P^{(t)}$ is the amount by which the agreement constraints are violated,

$$r_P^{(t)} = \frac{\sum_{(i,\alpha)\in E} \|\mathbf{M}_{i\alpha}\boldsymbol{q}_\alpha^{(t)} - \boldsymbol{p}_i^{(t)}\|^2}{\sum_{(i,\alpha)\in E} |\mathcal{Y}_i|} \in [0,1],$$

where the constant in the denominator ensures that $r_P^{(t)} \in [0,1]$. The *dual residual* $r_D^{(t)}$,

$$r_D^{(t)} = \frac{\sum_{(i,\alpha)\in E} \|\boldsymbol{p}_i^{(t)} - \boldsymbol{p}_i^{(t-1)}\|^2}{\sum_{(i,\alpha)\in E} |\mathcal{Y}_i|} \in [0,1],$$

is the amount by which a dual optimality condition is violated (see Boyd et al. 2011, §3.3 for details). We adopt as stopping criterion that these two residuals fall below a threshold, e.g., 10^{-6}.

4.4.2 Approximate Solutions of the Local Subproblems

The next proposition states that convergence may still hold if the local subproblems are only solved approximately. The importance of this result will be clear in Section 6, where we describe a general iterative algorithm for solving the local quadratic subproblems. Essentially, Proposition 9 allows these subproblems to be solved numerically up to some accuracy without compromising global convergence, as long as the accuracy of the solutions improves sufficiently fast over AD^3 iterations.

Proposition 9 (Eckstein and Bertsekas, 1992) *Let $\eta_t = \eta$, and for each iteration t, let $\widehat{\boldsymbol{q}}^{(t)}$ contain the exact solutions of* (18)*, and $\tilde{\boldsymbol{q}}^{(t)}$ those produced by an approximate algorithm. Then Proposition 7 still holds, provided that the sequence of errors is summable, i.e., $\sum_{t=1}^{\infty} \|\widehat{\boldsymbol{q}}^{(t)} - \tilde{\boldsymbol{q}}^{(t)}\| < \infty$.*

4.4.3 Runtime and Caching Strategies

In practice, considerable speed-ups can be achieved by caching the subproblems, a strategy which has also been proposed for the PSDD algorithm by Koo et al. (2010). After a few iterations, many variables $\boldsymbol{p}_i$ reach a consensus (i.e., $\boldsymbol{p}_i^{(t)} = \boldsymbol{q}_{i\alpha}^{(t+1)}, \forall \alpha \in \partial(i)$) and enter an idle state: they are left unchanged by the $\boldsymbol{p}$-update (line 10), and so do the Lagrange variables $\boldsymbol{\lambda}_{i\alpha}^{(t+1)}$ (line 11). If at iteration t all variables in a subproblem at factor α are idle, then $\boldsymbol{q}_\alpha^{(t+1)} = \boldsymbol{q}_\alpha^{(t)}$, hence the corresponding subproblem does not need to be solved. Typically, many variables and subproblems enter this idle state after the first few rounds. We will show the practical benefits of caching in the experimental section (Section 7.4, Figure 9).

4. This is particularly useful if inference is embedded in learning, where it is more important to obtain a *fractional* solution of the relaxed primal than an approximate integer one (Kulesza and Pereira, 2007; Martins et al., 2009).

4.5 Exact Inference with Branch-and-Bound

Recall that AD^3, as just described, solves the LP-MAP *relaxation* of the actual problem. In some problems, this relaxation is tight (in which case a certificate of optimality will be obtained), but this is not always the case. When a fractional solution is obtained, it is desirable to have a strategy to recover the exact MAP solution.

Two observations are noteworthy. First, as we saw in Section 2.3, the optimal value of the relaxed problem LP-MAP provides an upper bound to the original problem MAP. In particular, any feasible dual point provides an upper bound to the original problem's optimal value. Second, during execution of the AD^3 algorithm, we always keep track of a sequence of feasible dual points (as guaranteed by Proposition 7, item 3. Therefore, each iteration constructs tighter and tighter upper bounds. In recent work (Das et al., 2012), we proposed a *branch-and-bound search* procedure that finds the exact solution of the ILP. The procedure works recursively as follows:

1. Initialize $L = -\infty$ (our best value so far).
2. Run Algorithm 2. If the solution $\boldsymbol{p}^*$ is integer, return $\boldsymbol{p}^*$ and set L to the objective value. If along the execution we obtain an upper bound less than L, then Algorithm 2 can be safely stopped and return "infeasible"—this is the *bound* part. Otherwise (if $\boldsymbol{p}^*$ is fractional) go to step 3.
3. Find the "most fractional" component of $\boldsymbol{p}^*$ (call it $p_j^*(.)$) and *branch*: for every $y_j \in \mathcal{Y}_j$, create a branch where $p_j(y_j) = 1$ and $p_j(y'_j) = 0$ for $y'_j \neq y_j$, and go to step 2, eventually obtaining an integer solution $\boldsymbol{p}^*|_{y_j}$ or infeasibility. Return the $\boldsymbol{p}^* \in \{\boldsymbol{p}^*|_{y_j}\}_{y_j \in \mathcal{Y}_j}$ that yields the largest objective value.

Although this procedure has worst-case exponential runtime, in many problems for which the relaxations are near-exact it is found empirically very effective. We will see one example in Section 7.3.

5. Local Subproblems in AD^3

This section shows how to solve the AD^3 local subproblems (18) exactly and efficiently, in several cases, including Ising models and logic constraint factors. These results will be complemented in Section 6, where a new procedure to handle *arbitrary* factors widens the applicability of AD^3. By subtracting a constant, re-scaling, and flipping signs, problem (18) can be written more compactly as

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|\mathbf{M}\boldsymbol{q}_\alpha - \boldsymbol{a}\|^2 - \boldsymbol{b}^\top \boldsymbol{q}_\alpha \\ \text{with respect to} \quad & \boldsymbol{q}_\alpha \in \mathbb{R}^{|\mathcal{Y}_\alpha|} \\ \text{subject to} \quad & \mathbf{1}^\top \boldsymbol{q}_\alpha = 1, \quad \boldsymbol{q}_\alpha \geq \mathbf{0}, \end{aligned} \tag{20}$$

where $\boldsymbol{a} := (\boldsymbol{a}_i)_{i \in \partial(\alpha)}$, with $\boldsymbol{a}_i := \boldsymbol{p}_i + \eta^{-1}(\boldsymbol{\theta}_{i\alpha} + \boldsymbol{\lambda}_{i\alpha})$; $\boldsymbol{b} := \eta^{-1}\boldsymbol{\theta}_\alpha$; and $\mathbf{M} := (\mathbf{M}_{i\alpha})_{i \in \partial(\alpha)}$ denotes a matrix with $\sum_i |\mathcal{Y}_i|$ rows and $|\mathcal{Y}_\alpha|$ columns.

We show that problem (20) has a closed-form solution or can be solved exactly and efficiently, in several cases; e.g., for Ising models, for factor graphs imposing first-order logic

(FOL) constraints, and for Potts models (after binarization). In these cases, AD^3 and the PSDD algorithm have (asymptotically) the same computational cost per iteration, up to a logarithmic factor.

5.1 Ising Models

Ising models are factor graphs containing only binary pairwise factors. A binary pairwise factor (say, α) is one connecting two binary variables (say, Y_1 and Y_2); thus $\mathcal{Y}_1 = \mathcal{Y}_2 = \{0, 1\}$ and $\mathcal{Y}_\alpha = \{00, 01, 10, 11\}$. Given that $\boldsymbol{q}_{1\alpha}, \boldsymbol{q}_{2\alpha} \in \Delta^2$, we can write $\boldsymbol{q}_{1\alpha} = (1 - z_1, z_1)$, $\boldsymbol{q}_{2\alpha} = (1 - z_2, z_2)$. Furthermore, since $\boldsymbol{q}_\alpha \in \Delta^4$ and marginalization requires that $q_\alpha(1,1) + q_\alpha(1,0) = z_1$ and $q_\alpha(0,1) + q_\alpha(1,1) = z_2$, we can also write $\boldsymbol{q}_\alpha = (1 - z_1 - z_2 + z_{12}, z_1 - z_{12}, z_2 - z_{12}, z_{12})$. Using this parameterization, problem (20) reduces to:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}(z_1 - c_1)^2 + \frac{1}{2}(z_2 - c_2)^2 - c_{12} z_{12} \\ \text{with respect to} \quad & z_1, z_2, z_{12} \in [0,1]^3 \\ \text{subject to} \quad & z_{12} \leq z_1, \quad z_{12} \leq z_2, \quad z_{12} \geq z_1 + z_2 - 1, \end{aligned} \tag{21}$$

where

$$\begin{aligned} c_1 &= \frac{a_{1\alpha}(1) + 1 - a_{1\alpha}(0) - b_\alpha(0,0) + b_\alpha(1,0)}{2} \\ c_2 &= \frac{a_{2\alpha}(1) + 1 - a_{2\alpha}(0) - b_\alpha(0,0) + b_\alpha(0,1)}{2} \\ c_{12} &= \frac{b_\alpha(0,0) - b_\alpha(1,0) - b_\alpha(0,1) + b_\alpha(1,1)}{2}. \end{aligned}$$

The next proposition (proved in Appendix B.1) establishes a closed form solution for this problem, which immediately translates into a procedure for SolveQP for binary pairwise factors.

Proposition 10 *Let $[x]_{\mathbb{U}} := \min\{\max\{x, 0\}, 1\}$ denote projection (clipping) onto the unit interval $\mathbb{U} := [0, 1]$. The solution (z_1^*, z_2^*, z_{12}^*) of problem (21) is the following. If $c_{12} \geq 0$,*

$$\begin{aligned} (z_1^*, z_2^*) &= \begin{cases} ([c_1]_{\mathbb{U}}, & [c_2 + c_{12}]_{\mathbb{U}}), & \text{if } c_1 > c_2 + c_{12} \\ ([c_1 + c_{12}]_{\mathbb{U}}, & [c_2]_{\mathbb{U}}), & \text{if } c_2 > c_1 + c_{12} \\ ([(c_1 + c_2 + c_{12})/2]_{\mathbb{U}}, & [(c_1 + c_2 + c_{12})/2]_{\mathbb{U}}), & \text{otherwise,} \end{cases} \\ z_{12}^* &= \min\{z_1^*, z_2^*\}; \end{aligned} \tag{22}$$

otherwise ,

$$\begin{aligned} (z_1^*, z_2^*) &= \begin{cases} ([c_1 + c_{12}]_{\mathbb{U}}, & [c_2 + c_{12}]_{\mathbb{U}}), & \text{if } c_1 + c_2 + 2c_{12} > 1 \\ ([c_1]_{\mathbb{U}}, & [c_2]_{\mathbb{U}}), & \text{if } c_1 + c_2 < 1 \\ ([(c_1 + 1 - c_2)/2]_{\mathbb{U}}, & [(c_2 + 1 - c_1)/2]_{\mathbb{U}}), & \text{otherwise,} \end{cases} \\ z_{12}^* &= \max\{0, z_1^* + z_2^* - 1\}. \end{aligned} \tag{23}$$

5.2 Factor Graphs with First-Order Logic Constraints

Hard constraint factors allow specifying "forbidden" configurations, and have been used in error-correcting decoders (Richardson and Urbanke, 2008), bipartite graph matching (Duchi et al., 2007), computer vision (Nowozin and Lampert, 2009), and natural language processing (Smith and Eisner, 2008). In many applications, *declarative constraints* are useful for injecting domain knowledge, and first-order logic (FOL) provides a natural language to express such constraints. This is particularly useful in learning from scarce annotated data (Roth and Yih, 2004; Punyakanok et al., 2005; Richardson and Domingos, 2006; Chang et al., 2008; Poon and Domingos, 2009).

In this section, we consider hard constraint factors linked to binary variables, with log-potential functions of the form

$$\theta_\alpha(\boldsymbol{y}_\alpha) = \begin{cases} 0, & \text{if } \boldsymbol{y}_\alpha \in \mathcal{S}_\alpha \\ -\infty, & \text{otherwise,} \end{cases}$$

where $\mathcal{S}_\alpha \subseteq \{0,1\}^{|\partial(\alpha)|}$ is an *acceptance set*. These factors can be used for imposing FOL constraints, as we describe next. We define the *marginal polytope* $\mathcal{Z}_\alpha$ of a hard constraint factor α as the convex hull of its acceptance set,

$$\mathcal{Z}_\alpha = \operatorname{conv} \mathcal{S}_\alpha. \tag{24}$$

As shown in Appendix B.2, the AD^3 subproblem (20) associated with a hard constraint factor is equivalent to that of computing an Euclidean projection onto its marginal polytope:

$$\begin{aligned} \text{minimize} \quad & \|\boldsymbol{z} - \boldsymbol{z}_0\|^2 \\ \text{with respect to} \quad & \boldsymbol{z} \in \mathcal{Z}_\alpha, \end{aligned} \tag{25}$$

where $z_{0i} := (a_i(1) + 1 - a_i(0))/2$, for $i \in \partial(\alpha)$. We now show how to compute this projection for several hard constraint factors that are building blocks for writing FOL constraints. Each of these factors performs a logical function, and hence we represent them graphically as *logic gates* (Figure 3).

5.2.1 One-Hot XOR (Uniqueness Quantification)

The "one-hot XOR" factor linked to $K \geq 1$ binary variables is defined through the following potential function:

$$\theta_{\text{XOR}}(y_1, \ldots, y_K) := \begin{cases} 0 & \text{if } \exists! k \in \{1, \ldots, K\} \text{ s.t. } y_k = 1 \\ -\infty & \text{otherwise,} \end{cases}$$

where $\exists!$ denotes "there is one and only one." The name "one-hot XOR" stems from the following fact: for $K = 2$, $\exp(\theta_{\text{XOR}}(.))$ is the logic eXclusive-OR function; the prefix "one-hot" expresses that this generalization to $K > 2$ only accepts configurations with precisely one "active" input (not to be mistaken with other XOR generalizations commonly used for parity checks). The XOR factor can be used for binarizing a categorical variable, and to express a statement in FOL of the form $\exists! x : R(x)$.

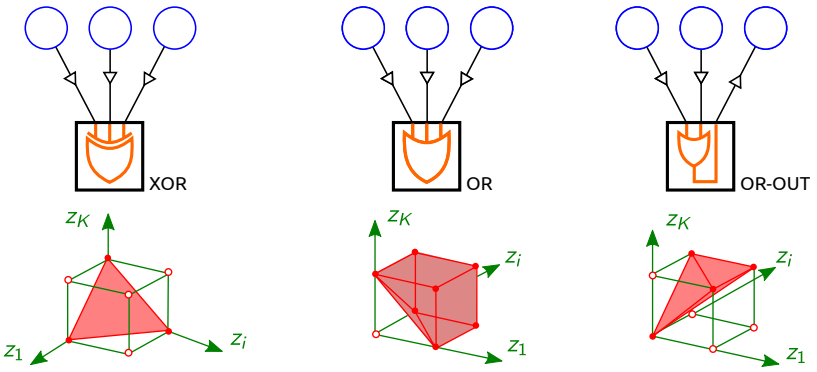


Figure 3: Logic factors and their marginal polytopes; the AD^3 subproblems (25) are projections onto these polytopes. Left: the one-hot XOR factor (its marginal polytope is the probability simplex). Middle: the OR factor. Right: the OR-with-output factor.

From (24), the marginal polytope associated with the one-hot XOR factor is

$$\mathcal{Z}_{\text{XOR}} = \text{conv}\left\{\boldsymbol{y} \in \{0,1\}^K \mid \exists! k \in \{1,\dots,K\} \text{ s.t. } y_k = 1\right\} = \Delta^K$$

as illustrated in Figure 3. Therefore, the AD^3 subproblem for the XOR factor consists in projecting onto the probability simplex, a problem well studied in the literature (Brucker, 1984; Michelot, 1986; Duchi et al., 2008). In Appendix B.3, we describe a simple $O(K \log K)$ algorithm. Note that there are $O(K)$ algorithms for this problem which are slightly more involved.

5.2.2 OR (Existential Quantification)

This factor represents a disjunction of $K \geq 1$ binary variables,

$$\theta_{\text{OR}}(y_1,\dots,y_K) := \begin{cases} 0 & \text{if } \exists k \in \{1,\dots,K\} \text{ s.t. } y_k = 1 \\ -\infty & \text{otherwise,} \end{cases}$$

The OR factor can be used to represent a statement in FOL of the form $\exists x : R(x)$.

From Proposition 16, the marginal polytope associated with the OR factor is:

$$\begin{aligned} \mathcal{Z}_{\text{OR}} &= \text{conv}\left\{\boldsymbol{y} \in \{0,1\}^K \mid \exists k \in \{1,\dots,K\} \text{ s.t. } y_k = 1\right\} \\ &= \left\{\boldsymbol{z} \in [0,1]^K \;\middle|\; \sum_{k=1}^K z_k \geq 1\right\}; \end{aligned}$$

geometrically, it is a "truncated" hypercube, as depicted in Figure 3. We derive a $O(K \log K)$ algorithm for projecting onto $\mathcal{Z}_{\text{OR}}$, using a sifting technique and a sort operation (see Appendix B.4).

5.2.3 Logical Variable Assignments: OR-With-Output

The two factors above define a constraint on a group of existing variables. Alternatively, we may want to define a new variable (say, y_{K+1}) which is the result of an operation involving other variables (say, $y_1, \ldots, y_K$). Among other things, this will allow dealing with "soft constraints," i.e., constraints that can be violated but whose violation will decrease the score by some penalty. An example is the OR-with-output factor:

$$\theta_{\text{OR}-\text{out}}(y_1, \ldots, y_K, y_{K+1}) := \begin{cases} 1 & \text{if } y_{K+1} = y_1 \vee \cdots \vee y_K \\ 0 & \text{otherwise.} \end{cases}$$

This factor constrains the variable y_{K+1} to indicate the existence of one or more active variables among $y_1, \ldots, y_K$. It can be used to express the following statement in FOL: $T(x) := \exists z : R(x, z)$.

The marginal polytope associated with the OR-with-output factor (also depicted in Figure 3):

$$\begin{aligned} \mathcal{Z}_{\text{OR}-\text{out}} &= \text{conv}\left\{ \boldsymbol{y} \in \{0,1\}^{K+1} \;\middle|\; y_{K+1} = y_1 \vee \cdots \vee y_K \right\} \\ &= \left\{ \boldsymbol{z} \in [0,1]^{K+1} \;\middle|\; \sum_{k=1}^{K} z_k \geq z_{K+1},\ z_k \leq z_{K+1}, \forall k \in \{1, \ldots, K\} \right\}. \end{aligned}$$

Although projecting onto $\mathcal{Z}_{\text{OR}-\text{out}}$ is slightly more complicated than the previous cases, in Appendix B.5, we propose (and prove correctness of) an $O(K \log K)$ algorithm for this task.

5.2.4 Negations, De Morgan's Laws, and AND-With-Output

The three factors just presented can be extended to accommodate *negated* inputs, thus adding flexibility. Solving the corresponding AD^3 subproblems can be easily done by reusing the methods that solve the original problems. For example, it is straightforward to handle negated conjunctions (NAND),

$$\begin{aligned} \theta_{\text{NAND}}(y_1, \ldots, y_K) &:= \begin{cases} -\infty & \text{if } y_k = 1,\ \forall k \in \{1, \ldots, K\} \\ 0 & \text{otherwise,} \end{cases} \\ &= \theta_{\text{OR}}(\neg y_1, \ldots, \neg y_K), \end{aligned}$$

as well as implications (IMPLY),

$$\begin{aligned} \theta_{\text{IMPLY}}(y_1, \ldots, y_K, y_{K+1}) &:= \begin{cases} 0 & \text{if } (y_1 \wedge \cdots \wedge y_K) \Rightarrow y_{K+1} \\ -\infty & \text{otherwise} \end{cases} \\ &= \theta_{\text{OR}}(\neg y_1, \ldots, \neg y_K, y_{K+1}). \end{aligned}$$

In fact, from De Morgan's laws, $\neg\left(Q_1(x) \wedge \cdots \wedge Q_K(x)\right)$ is equivalent to $\neg Q_1(x) \vee \cdots \vee \neg Q_K(x)$, and $(Q_1(x) \wedge \cdots \wedge Q_K(x)) \Rightarrow R(x)$ is equivalent to $(\neg Q_1(x) \vee \cdots \vee \neg Q_K(x)) \vee R(x)$. Another example is the AND-with-output factor,

$$\begin{aligned} \theta_{\text{AND}-\text{out}}(y_1, \ldots, y_K, y_{K+1}) &:= \begin{cases} 0 & \text{if } y_{K+1} = y_1 \wedge \cdots \wedge y_K \\ -\infty & \text{otherwise} \end{cases} \\ &= \theta_{\text{OR}-\text{out}}(\neg y_1, \ldots, \neg y_K, \neg y_{K+1}), \end{aligned}$$

which can be used to impose FOL statements of the form $T(x) := \forall z : R(x, z)$.

Let α be a binary constraint factor with marginal polytope $\mathcal{Z}_\alpha$, and β a factor obtained from α by negating the kth variable. For notational convenience, let $\mathrm{sym}_k : [0,1]^K \to [0,1]^K$ be defined as $(\mathrm{sym}_k(\boldsymbol{z}))_k = 1 - z_k$ and $(\mathrm{sym}_k(\boldsymbol{z}))_i = z_i$, for $i \neq k$. Then, the marginal polytope $\mathcal{Z}_\beta$ is a symmetric transformation of $\mathcal{Z}_\alpha$,

$$\mathcal{Z}_\beta \;=\; \Big\{ \boldsymbol{z} \in [0,1]^K \;\Big|\; \mathrm{sym}_k(\boldsymbol{z}) \in \mathcal{Z}_\alpha \Big\},$$

and, if $\mathrm{proj}_{\mathcal{Z}_\alpha}$ denotes the projection operator onto $\mathcal{Z}_\alpha$,

$$\mathrm{proj}_{\mathcal{Z}_\beta}(\boldsymbol{z}) = \mathrm{sym}_k \left(\mathrm{proj}_{\mathcal{Z}_\alpha}(\mathrm{sym}_k(\boldsymbol{z})) \right).$$

Naturally, $\mathrm{proj}_{\mathcal{Z}_\beta}$ can be computed as efficiently as $\mathrm{proj}_{\mathcal{Z}_\alpha}$ and, by induction, this procedure can be generalized to an arbitrary number of negated variables.

5.3 Potts Models and Graph Binarization

Although general factors lack closed-form solutions of the corresponding AD^3 subproblem (20), it is possible to *binarize* the graph, i.e., to convert it into an equivalent one that only contains binary variables and XOR factors. The procedure is as follows:

- For each variable node $i \in V$, define binary variables $U_{i,y_i} \in \{0,1\}$, for each state $y_i \in \mathcal{Y}_i$; link these variables to a XOR factor, imposing $\sum_{y_i \in \mathcal{Y}_i} p_i(y_i) = 1$.
- For each factor $\alpha \in F$, define binary variables $U_{\alpha,\boldsymbol{y}_\alpha} \in \{0,1\}$ for every $\boldsymbol{y}_\alpha \in \mathcal{Y}_\alpha$. For each edge $(i,\alpha) \in E$ and each $y_i \in \mathcal{Y}_i$, link variables $\{U_{\alpha,\boldsymbol{y}_\alpha} \mid \boldsymbol{y}_\alpha \sim y_i\}$ and $\neg U_{i,y_i}$ to a XOR factor; this imposes the constraint $p_i(y_i) = \sum_{\boldsymbol{y}_\alpha \sim y_i} q_\alpha(\boldsymbol{y}_\alpha)$.

The resulting binary graph is one for which we already presented the machinery needed for solving efficiently the corresponding AD^3 subproblems. As an example, for Potts models (graphs with only pairwise factors and variables that have more than two states), the computational cost per AD^3 iteration on the binarized graph is asymptotically the same as that of the PSDD and other message-passing algorithms; for details, see Martins (2012).

6. An Active Set Method For Solving the AD^3 Subproblems

In this section, we complement the results of Section 5 with a general *active-set procedure* for solving the AD^3 subproblems for *arbitrary* factors, the only requirement being a black-box MAP solver—the same as the PSDD algorithm. This makes AD^3 applicable to a wide range of problems. In particular, it makes possible to handle *structured factors*, by invoking specialized MAP decoders (functions ComputeMAP in Algorithm 1). In practice, as we will see in Section 7, the active set method we next present largely outperforms the graph binarization strategy outlined in Section 5.3.

Our active set method is based on Nocedal and Wright (1999, Section 16.4); it is an iterative algorithm that addresses the AD^3 subproblems (20) by solving a sequence of linear problems. The next crucial proposition (proved in Appendix C) states that the problem (20) always admits a *sparse solution*.

Proposition 11 *Problem* (20) *admits a solution* $\boldsymbol{q}_\alpha^* \in \mathbb{R}^{|\mathcal{Y}_\alpha|}$ *with at most* $\sum_{i\in\partial(\alpha)} |\mathcal{Y}_i| - |\partial(\alpha)| + 1$ *non-zero components.*

The fact that the solution lies in a low dimensional subspace makes active set methods appealing, since they only keep track of an *active set* of variables, that is, the non-zero components of $\boldsymbol{q}_\alpha$. Proposition 11 tells us that such an algorithm only needs to maintain at most $O(\sum_i |\mathcal{Y}_i|)$ elements in the active set—note the *additive*, rather than multiplicative, dependency on the number of values of the variables. Our active set method seeks to identify the low-dimensional support of the solution $\boldsymbol{q}_\alpha^*$, by generating sparse iterates $\boldsymbol{q}_\alpha^{(1)}, \boldsymbol{q}_\alpha^{(2)}, \ldots$, while it maintains a working set $W \subseteq \mathcal{Y}_\alpha$ with the inequality constraints of (20) that are *inactive* along the way (i.e., those $\boldsymbol{y}_\alpha$ for which $q_\alpha(\boldsymbol{y}_\alpha) > 0$ holds strictly). Each iteration adds or removes elements from the working set while it monotonically decreases the objective of (20).[5]

Lagrangian and KKT conditions. Let τ and $\boldsymbol{\mu}$ be dual variables associated with the equality and inequality constraints of (20), respectively. The Lagrangian function is

$$L(\boldsymbol{q}_\alpha, \tau, \boldsymbol{\mu}) = \frac{1}{2}\|\mathbf{M}\boldsymbol{q}_\alpha - \boldsymbol{a}\|^2 - \boldsymbol{b}^\top \boldsymbol{q}_\alpha - \tau(1 - \mathbf{1}^\top \boldsymbol{q}_\alpha) - \boldsymbol{\mu}^\top \boldsymbol{q}_\alpha.$$

This gives rise to the following Karush-Kuhn-Tucker (KKT) conditions:

$$\mathbf{M}^\top(\boldsymbol{a} - \mathbf{M}\boldsymbol{q}_\alpha) + \boldsymbol{b} = \tau\mathbf{1} - \boldsymbol{\mu} \quad (\nabla_{\boldsymbol{q}_\alpha} L = \mathbf{0}) \tag{26}$$

$$\mathbf{1}^\top \boldsymbol{q}_\alpha = 1,\ \boldsymbol{q}_\alpha \geq \mathbf{0},\ \boldsymbol{\mu} \geq \mathbf{0} \quad \text{(Primal/dual feasibility)} \tag{27}$$

$$\boldsymbol{\mu}^\top \boldsymbol{q}_\alpha = \mathbf{0} \quad \text{(Complementary slackness).} \tag{28}$$

The method works at follows. At each iteration s, it first checks if the current iterate $\boldsymbol{q}_\alpha^{(s)}$ is a *subspace minimizer*, i.e., if it optimizes the objective of (20) in the sparse subspace defined by the working set W, $\{\boldsymbol{q}_\alpha \in \Delta^{|\mathcal{Y}_\alpha|} \mid q_\alpha(\boldsymbol{y}_\alpha) = 0, \forall \boldsymbol{y}_\alpha \notin W\}$. This check can be made by first solving a relaxation where the inequality constraints are ignored. Since in this subspace the components of $\boldsymbol{q}_\alpha$ not in W will be zeros, one can simply delete those entries from $\boldsymbol{q}_\alpha$ and $\boldsymbol{b}$ and the corresponding columns in $\mathbf{M}$; we use a horizontal bar to denote these truncated $\mathbb{R}^{|W|}$-vectors. The problem can be written as:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|\bar{\mathbf{M}}\bar{\boldsymbol{q}}_\alpha - \boldsymbol{a}\|^2 - \bar{\boldsymbol{b}}^\top \bar{\boldsymbol{q}}_\alpha \\ \text{with respect to} \quad & \bar{\boldsymbol{q}}_\alpha \in \mathbb{R}^{|W|} \\ \text{subject to} \quad & \mathbf{1}^\top \bar{\boldsymbol{q}}_\alpha = 1. \end{aligned} \tag{29}$$

The solution of this equality-constrained QP can be found by solving a system of KKT equations:[6]

$$\begin{bmatrix} \bar{\mathbf{M}}^\top \bar{\mathbf{M}} & \mathbf{1} \\ \mathbf{1}^\top & 0 \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{q}}_\alpha \\ \tau \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{M}}^\top \boldsymbol{a} + \bar{\boldsymbol{b}} \\ 1 \end{bmatrix}. \tag{30}$$

5. Our description differs from Nocedal and Wright (1999) in which their working set contains *active* constraints rather than the inactive ones. In our case, most constraints are active for the optimal $\boldsymbol{q}_\alpha^*$, therefore it is appealing to store the ones that are not.

6. Note that this is a low-dimensional problem, since we are working in a sparse working set. By caching the inverse of the matrix in the left-hand side, this system can be solved in time $O(|W|^2)$ at each iteration.

The solution of (30) will give $(\widehat{\boldsymbol{q}}_\alpha, \widehat{\tau})$, where $\widehat{\boldsymbol{q}}_\alpha \in \mathbb{R}^{|\mathcal{Y}_\alpha|}$ is padded back with zeros. If it happens that $\widehat{\boldsymbol{q}}_\alpha = \boldsymbol{q}_\alpha^{(s)}$, then this means that the current iterate $\boldsymbol{q}_\alpha^{(s)}$ is a subspace minimizer; otherwise a new iterate $\boldsymbol{q}_\alpha^{(s+1)}$ will be computed. We next discuss these two events.

- *Case 1: $\boldsymbol{q}_\alpha^{(s)}$ is a subspace minimizer.* If this happens, then it may be the case that $\boldsymbol{q}_\alpha^{(s)}$ is the optimal solution of (20). By looking at the KKT conditions (26)–(28), we have that this will happen iff $\mathbf{M}^\top(\boldsymbol{a} - \mathbf{M}\boldsymbol{q}_\alpha^{(s)}) + \boldsymbol{b} \leq \tau^{(s)}\mathbf{1}$. Define $\boldsymbol{w} := \boldsymbol{a} - \mathbf{M}\boldsymbol{q}_\alpha$. The condition above is equivalent to

$$\max_{\boldsymbol{y}_\alpha \in \mathcal{Y}_\alpha} \left(b(\boldsymbol{y}_\alpha) + \sum_{i \in \partial(\alpha)} w_i(y_i) \right) \leq \tau^{(s)}.$$

  It turns out that this maximization is precisely a *local MAP inference problem*, given a vector of unary potentials $\boldsymbol{w}$ and factor potentials $\boldsymbol{b}$. Thus, the maximizer $\widehat{\boldsymbol{y}}_\alpha$ can be computed via the ComputeMAP procedure, which we assume available. If $b(\widehat{\boldsymbol{y}}_\alpha) + \sum_{i \in \partial(\alpha)} w_i(\widehat{y}_i) \leq \tau^{(s)}$, then the KKT conditions are satisfied and we are done. Otherwise, $\widehat{\boldsymbol{y}}_\alpha$ indicates the most violated condition; we will add it to the active set W, and proceed.

- *Case 2: $\boldsymbol{q}_\alpha^{(s)}$ is not a subspace minimizer.* If this happens, then we compute a new iterate $\boldsymbol{q}_\alpha^{(s+1)}$ by keeping searching in the same subspace. We have already solved a relaxation in (29). If we have $\widehat{q}_\alpha(\boldsymbol{y}_\alpha) \geq 0$ for all $\boldsymbol{y}_\alpha \in W$, then the relaxation is tight, so we just set $\boldsymbol{q}_\alpha^{(s+1)} := \widehat{\boldsymbol{q}}_\alpha$ and proceed. Otherwise, we move as much as possible in the direction of $\widehat{\boldsymbol{q}}_\alpha$ while keeping feasibility, by defining $\boldsymbol{q}_\alpha^{(s+1)} := (1-\beta)\boldsymbol{q}_\alpha^{(s)} + \beta\widehat{\boldsymbol{q}}_\alpha$—as described in Nocedal and Wright (1999), the value of $\beta \in [0, 1]$ can be computed in closed form:

$$\beta = \min \left\{ 1, \min_{\boldsymbol{y}_\alpha \in W \,:\, q_\alpha^{(s)}(\boldsymbol{y}_\alpha) > \widehat{q}_\alpha(\boldsymbol{y}_\alpha)} \frac{q_\alpha^{(s)}(\boldsymbol{y}_\alpha)}{q_\alpha^{(s)}(\boldsymbol{y}_\alpha) - \widehat{q}_\alpha(\boldsymbol{y}_\alpha)} \right\}. \tag{31}$$

  If $\beta < 1$, this update will have the effect of making one of the constraints active, by zeroing out $q_\alpha^{(s+1)}(\boldsymbol{y}_\alpha)$ for the minimizing $\boldsymbol{y}_\alpha$ above. This so-called "blocking constraint" is thus be removed from the working set W.

Algorithm 3 describes the complete procedure. The active set W is initialized arbitrarily: a strategy that works well in practice is, in the first AD^3 iteration, initialize $W := \{\widehat{\boldsymbol{y}}_\alpha\}$, where $\widehat{\boldsymbol{y}}_\alpha$ is the MAP configuration given log-potentials $\boldsymbol{a}$ and $\boldsymbol{b}$; and in subsequent AD^3 iterations, warm-start W with the support of the solution obtained in the previous iteration.

Each iteration of Algorithm 3 improves the objective of (20), and, with a suitable strategy to prevent cycles and stalling, the algorithm is guaranteed to stop after a finite

Note also that adding a new configuration $\boldsymbol{y}_\alpha$ to the active set, corresponds to inserting a new column in $\bar{\mathbf{M}}$, thus the matrix inversion requires updating $\bar{\mathbf{M}}^\top\bar{\mathbf{M}}$. From the definition of $\mathbf{M}$ and simple algebra, the $(\boldsymbol{y}_\alpha, \boldsymbol{y}'_\alpha)$ entry in $\mathbf{M}^\top\mathbf{M}$ is simply the *number of common values* between the configurations $\boldsymbol{y}_\alpha$ and $\boldsymbol{y}'_\alpha$. Hence, when a new configuration $\boldsymbol{y}_\alpha$ is added to the active set W, that configuration needs to be compared with all the others already in W.

Algorithm 3 Active Set Algorithm for Solving a General AD^3 Subproblem

input: Parameters $\boldsymbol{a}, \boldsymbol{b}, \mathbf{M}$, starting point $\boldsymbol{q}_\alpha^{(0)}$
initialize $W^{(0)}$ as the support of $\boldsymbol{q}_\alpha^{(0)}$
for $s = 0, 1, 2, \ldots$ **do**
  solve the KKT system and obtain $\widehat{\boldsymbol{q}}_\alpha$ and $\widehat{\tau}$ (30)
  if $\widehat{\boldsymbol{q}}_\alpha = \boldsymbol{q}_\alpha^{(s)}$ **then**
    compute $\boldsymbol{w} := \boldsymbol{a} - \mathbf{M}\widehat{\boldsymbol{q}}_\alpha$
    obtain the tighter constraint $\widehat{\boldsymbol{y}}_\alpha$ via $\boldsymbol{e}_{\widehat{\boldsymbol{y}}_\alpha} = \textsc{ComputeMAP}(\boldsymbol{b} + \mathbf{M}^\top \boldsymbol{w})$
    if $b(\widehat{\boldsymbol{y}}_\alpha) + \sum_{i \in \partial(\alpha)} w_i(\widehat{y}_i) \leq \widehat{\tau}$ **then**
      return solution $\widehat{\boldsymbol{q}}_\alpha$
    else
      add the most violated constraint to the active set: $W^{(s+1)} := W^{(s)} \cup \{\widehat{\boldsymbol{y}}_\alpha\}$
    end if
  else
    compute the interpolation constant β as in (31)
    set $\boldsymbol{q}_\alpha^{(s+1)} := (1 - \beta)\boldsymbol{q}_\alpha^{(s)} + \beta\widehat{\boldsymbol{q}}_\alpha$
    if if $\beta < 1$ **then**
      pick the blocking constraint $\widehat{\boldsymbol{y}}_\alpha$ in (31)
      remove $\widehat{\boldsymbol{y}}_\alpha$ from the active set: $W^{(s+1)} := W^{(s)} \setminus \{\widehat{\boldsymbol{y}}_\alpha\}$
    end if
  end if
end for
output: $\widehat{\boldsymbol{q}}_\alpha$

number of steps (Nocedal and Wright, 1999, Theorem 16.5). In practice, since it is run as a subroutine of AD^3, Algorithm 3 does not need to be run to optimality, which is convenient in early iterations of AD^3 (this is supported by Proposition 9). The ability to warm-start with the solution from the previous round is very useful in practice: we have observed that, thanks to this warm-starting strategy, very few inner iterations are typically necessary for the correct active set to be identified. We will see some empirical evidence in Section 7.4.

7. Experiments

In this section, we provide an empirical comparison between AD^3 (Algorithm 2) and four other algorithms: generalized MPLP (Globerson and Jaakkola, 2008); norm-product BP (Hazan and Shashua, 2010);[7] the PSDD algorithm of Komodakis et al. (2007) (Algorithm 1) and its accelerated version introduced by Jojic et al. (2010). All these algorithms address the LP-MAP problem; the first are message-passing methods performing block coordinate descent in the dual, whereas the last two are based on dual decomposition. The norm-product BP and accelerated dual decomposition algorithms introduce a temperature parameter to smooth their dual objectives. All the baselines have the same algorithmic complexity per

7. For norm-product BP, we adapted the code provided by the authors, using the "trivial" counting numbers $c_\alpha = 1$, $c_{i\alpha} = 0$, and $c_i = 0$, $\forall (i, \alpha) \in E$, which leads to a concave entropy approximation.

iteration, which is asymptotically the same as that of the AD^3 applied to a binarized graph, but different from that of AD^3 with the active set method.

We compare the performance of the algorithms above in several data sets, including synthetic Ising and Potts models, protein design problems, and two problems in natural language processing: frame-semantic parsing and non-projective dependency parsing. The graphical models associated with these problems are quite diverse, containing pairwise binary factors (AD^3 subproblems solved as described in Section 5.1), first-order logic factors (addressed using the tools of Section 5.2), dense factors, and structured factors (tackled with the active set method of Section 6).

7.1 Synthetic Ising and Potts Models

We start by comparing AD^3 with their competitors on synthetic Ising and Potts models.

7.1.1 Ising Models

Figure 4 reports experiments with random Ising models, with single-node log-potentials chosen as $\theta_i(1) - \theta_i(0) \sim \mathcal{U}[-1, 1]$ and random edge couplings in $\mathcal{U}[-\rho, \rho]$, where $\rho \in \{0.1, 0.2, 0.5, 1.0\}$. Decompositions are edge-based for all methods. For MPLP and norm-product BP, primal feasible solutions $(\widehat{y}_i)_{i \in V}$ are obtained by decoding the single node messages (Globerson and Jaakkola, 2008); for the dual decomposition methods, $\widehat{y}_i = \operatorname{argmax}_{y_i} p_i(y_i)$.

We observe that PSDD is the slowest algorithm, taking a long time to find a "good" primal feasible solution, arguably due to the large number of components. The accelerated dual decomposition method (Jojic et al., 2010) is also not competitive in this setting, as it takes many iterations to reach a near-optimal region. MPLP, norm-product, and AD^3 all perform very similarly regarding convergence to the dual objective, with a slight advantage of the latter two. Regarding their ability to find a "good" feasible primal solution, AD^3 and norm-product BP seem to outperform their competitors. In a batch of 100 experiments using a coupling $\rho = 0.5$, AD^3 found a best dual than MPLP in 18 runs and it lost 11 times (the remaining 71 runs were ties); it won over norm-product BP 73 times and never lost. In terms of primal solutions, AD^3 won over MPLP in 47 runs and it lost 12 times (41 ties); and it won over norm-product in 49 runs and it lost 33 times (in all cases, relative differences lower than 1×10^{-6} were considered as ties).

7.1.2 Potts Models

The effectiveness of AD^3 in the non-binary case is assessed using random Potts models, with single-node log-potentials chosen as $\theta_i(y_i) \sim \mathcal{U}[-1, 1]$ and pairwise log-potentials as $\theta_{ij}(y_i, y_j) \sim \mathcal{U}[-10, 10]$ if $y_i = y_j$ and 0 otherwise. All the baselines use the same edge decomposition as before, since they handle multi-valued variables; for AD^3, we tried two variants: one where the graph is binarized (see Section 5.3); and one which works in the original graph through the active set method, as described in Section 6.

As shown in Figure 5, MPLP and norm-product BP decrease the objective very rapidly in the beginning and then slow down considerably; the accelerated dual decomposition algorithm, although slower in early iterations, eventually surpasses them. Both variants of AD^3 converge as fast as the accelerated dual decomposition algorithm in later iterations,

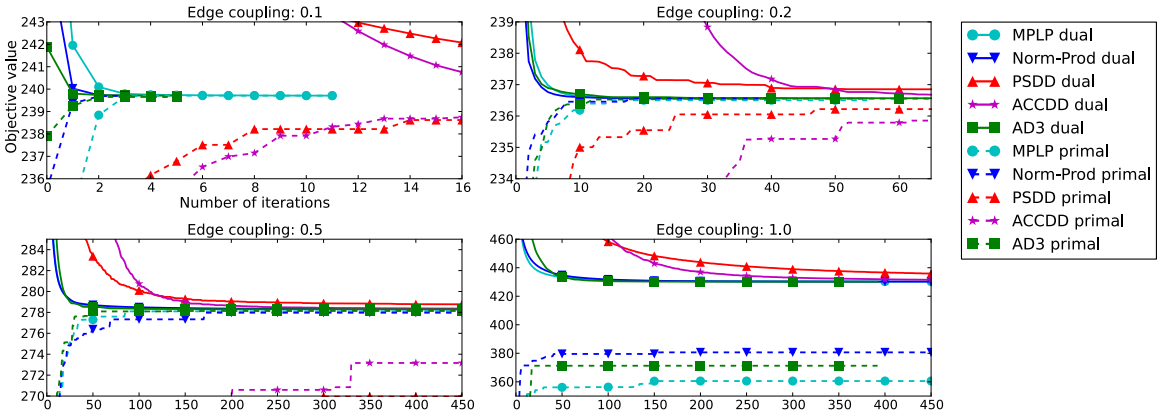


Figure 4: Evolution of the dual objective and the best primal feasible one in the experiments with 30×30 random Ising models, generated as described in the main text. For the subgradient method, the step sizes are $\eta_t = \eta_0/k(t)$, where $k(t)$ is the number of times the dual decreased up to the tth iteration, and η_0 was chosen with hindsight in $\{0.001, 0.01, 0.1, 10\}$ to yield the best dual objective. For accelerated dual decomposition, the most favorable parameter $\epsilon \in \{0.1, 1, 10, 100\}$ was chosen. For norm-product BP, the temperature was set as $\tau = 0.001$, and the dual objective is computed with zero temperature (which led to better upper bounds). AD^3 uses $\eta = 0.1$ for all runs.

and are almost as fast as MPLP and norm-product in early iterations, getting the best of both worlds. Comparing the two variants of AD^3, we observe that the active set variant clearly outperforms the binarization variant. Notice that since AD^3 with the active set method involves more computation per iteration, we plot the objective values with respect to the normalized number of oracle calls (which matches the number of iterations for the other methods).

7.2 Protein Design

We compare AD^3 with the MPLP implementation[8] of Sontag et al. (2008) in the benchmark protein design problems[9] of Yanover et al. (2006). In these problems, the input is a three-dimensional shape, and the goal is to find the most stable sequence of amino acids in that shape. The problems can be represented as pairwise factor graphs, whose variables correspond to the identity of amino acids and rotamer configurations, thus having hundreds of possible states. Figure 6 plots the evolution of the dual objective over runtime, for two of the largest problem instances, i.e., those with 3167 (1fbo) and 1163 (1kw4) factors. These plots are representative of the typical performance obtained in other instances. In both cases, MPLP steeply decreases the objective at early iterations, but then reaches a plateau

8. Available at `http://cs.nyu.edu/~dsontag/code`; that code includes a "tightening" procedure for retrieving the exact MAP, which we don't use, since we are interested in the LP-MAP relaxation (which is what AD^3 addresses).
9. Available at `http://www.jmlr.org/papers/volume7/yanover06a/`.

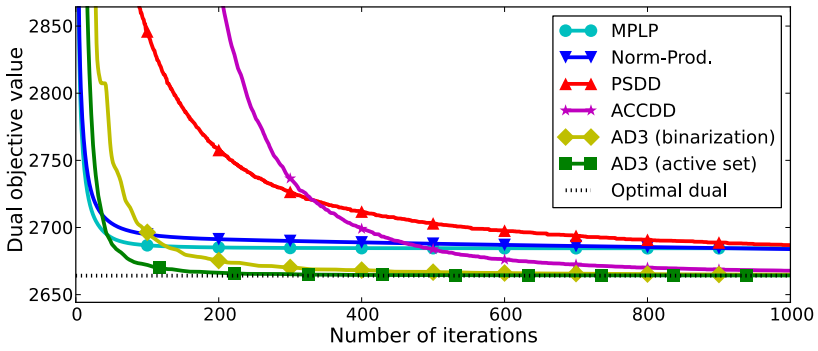


Figure 5: Evolution of the dual objective in the experiments with random 20×20 Potts models with 8-valued nodes, generated as described in the main text. For PSDD and the accelerated dual decomposition algorithm, we chose η_0 and ϵ as before. For AD^3, we set $\eta = 1.0$ in both settings (active set and binarization). In the active set method, no caching was used and the plotted number of iterations is corrected to make it comparable with the remaining algorithms, since each outer iteration of AD^3 requires several calls to a MAP oracle (we plot the normalized number of oracle calls instead). Yet, due to warm-starting, the average number of inner iterations is only 1.04, making the active set method extremely efficient. For all methods, the markers represent every 100 iterations.

with no further significant improvement. AD^3 rapidly surpasses MPLP in obtaining a better dual objective. Finally, observe that although earlier iterations of AD^3 take longer than those of MPLP, this cost is amortized in later iterations, by warm-starting the active set method.

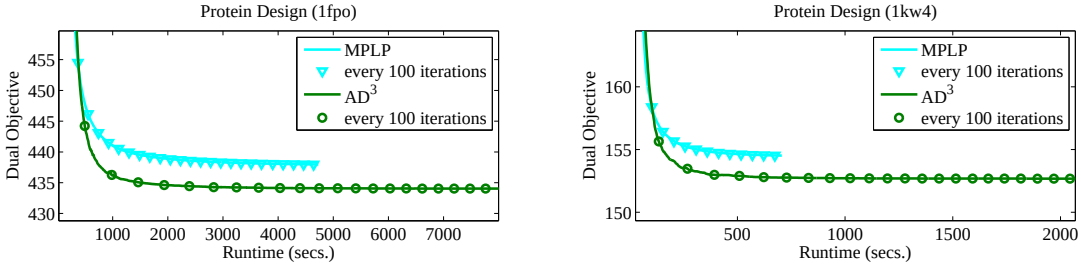


Figure 6: Protein design experiments (see main text for details). In AD^3, η is adjusted as proposed by Boyd et al. (2011, §3.4.1), initialized at $\eta = 1.0$ and the subproblems are solved by the proposed active set method. Although the plots are with respect to runtime, they also indicate iteration counts.

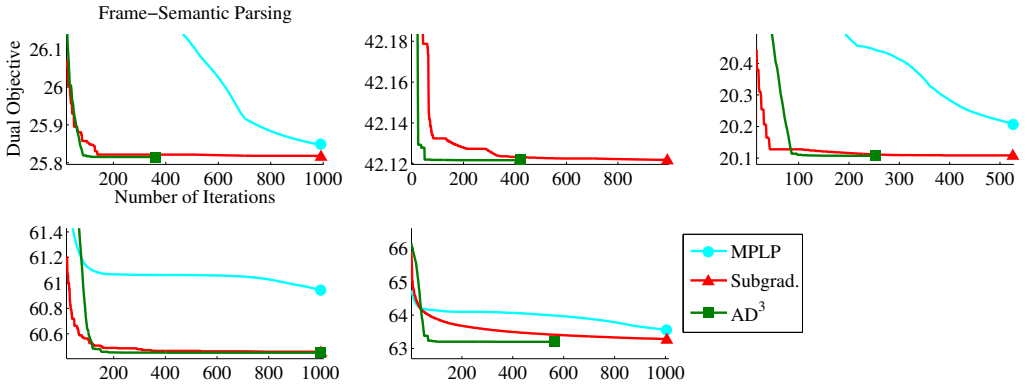


Figure 7: Experiments in five frame-semantic parsing problems (Das, 2012, Section 5.5). The projected subgradient uses $\eta_t = \eta_0/t$, with $\eta_0 = 1.0$ (found to be the best choice for all examples). In AD^3, η is adjusted as proposed by Boyd et al. (2011), initialized at $\eta = 1.0$.

7.3 Frame-Semantic Parsing

We now report experiments on a natural language processing task involving logic constraints: *frame-semantic parsing*, using the FrameNet lexicon (Fillmore, 1976). The goal is to predict the set of arguments and roles for a predicate word in a sentence, while respecting several constraints about the frames that can be evoked. The resulting graphical models are binary constrained factor graphs with FOL constraints (see Das et al. 2012 for details about this task). Figure 7 shows the results of AD^3, MPLP, and PSDD on the five most difficult problems (which have between 321 and 884 variables, and between 32 and 59 factors), the ones in which the LP relaxation is not tight. Unlike MPLP and PSDD, which did not converge after 1000 iterations, AD^3 achieves convergence in a few hundreds of iterations for all but one example. Since these examples have a fractional LP-MAP solution, we applied the branch-and-bound procedure described in Section 4.5 to obtain the exact MAP for these examples. The whole data set contains 4,462 instances, which were parsed by this exact variant of the AD^3 algorithm in only 4.78 seconds, against 43.12 seconds of CPLEX, a state-of-the-art commercial ILP solver.

7.4 Dependency Parsing

The final set of experiments assesses the ability of AD^3 to handle problems with structured factors. The task is *dependency parsing* (illustrated in the left part of Figure 8), an important problem in natural language processing (Eisner, 1996; McDonald et al., 2005), to which dual decomposition has been recently applied (Koo et al., 2010). We use an English data set derived from the Penn Treebank (PTB)(Marcus et al., 1993), converted to dependencies by applying the head rules of Yamada and Matsumoto (2003); we follow the common procedure of training in sections §02–21 (39,832 sentences), using §22 as validation data (1,700 sentences), and testing on §23 (2,416 sentences). We ran a part-of-speech tagger on the validation and test splits, and devised a linear model using various features depend-

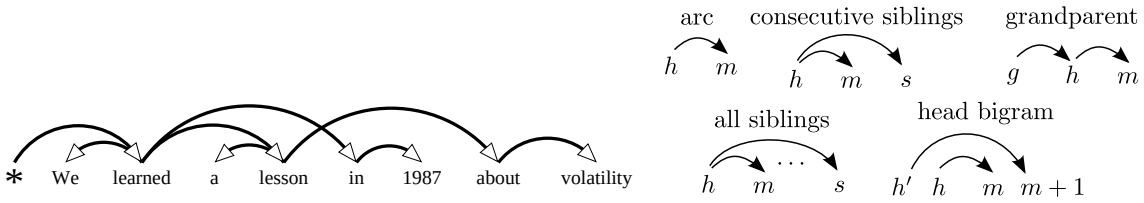


Figure 8: Left: example of a sentence (input) and its dependency parse tree (output to be predicted); this is a directed spanning tree where each arc (h, m) represent a syntactic relationships between a *head* word h and the a *modifier* word m. Right: the parts used in our models. *Arcs* are the basic parts: any dependency tree can be "read out" from its arcs. *Consecutive siblings* and *grandparent* parts introduce horizontal and vertical Markovization. We break the horizontal Markovianity via *all siblings* parts (which look at arbitrary pairs of siblings, not necessarily consecutive). Inspired by transition-based parsers, we also adopt *head bigram* parts, which look at the heads attached to consecutive words.

ing on words, part-of-speech tags, and arc direction and length. Our features decompose over the parts illustrated in the right part of Figure 8. We consider two different models in our experiments: a *second order model* with scores for arcs, consecutive siblings, and grandparents; a *full model*, which also has scores for arbitrary siblings and head bigrams.

If only scores for arcs were used, the problem of obtaining a parse tree with maximal score could be solved efficiently with a maximum directed spanning tree algorithm (Chu and Liu, 1965; Edmonds, 1967; McDonald et al., 2005); the addition of any of the other scores makes the problem NP-hard (McDonald and Satta, 2007). A factor graph representing the second order model, proposed by Smith and Eisner (2008) and Koo et al. (2010), contains binary variables representing the candidate arcs, a hard-constraint factor imposing the tree constraint, and head automata factors modeling the sequences of consecutive siblings and grandparents. The full model has additional binary pairwise factors for each possible pair of siblings (significantly increasing the number of factors), and a sequential factor modeling the sequence of heads.[10] We compare the PSDD and AD^3 algorithms for this task, using the decompositions above, which are the same for both methods. These decompositions select the largest factors for which efficient MAP oracles exist, based on the Chu-Liu-Edmonds algorithm and on dynamic programming. The active set method enables AD^3 to depend only on these MAP oracles.

Figure 9 illustrates the remarkable speed-ups that the caching and warm-starting procedures bring to both the AD^3 and PSDD algorithms. A similar conclusion was obtained by Koo et al. (2010) for PSDD and by Martins et al. (2011b) for AD^3 in a different factor graph. Figure 10 shows average runtimes for both algorithms, as a function of the sentence length, and plots the percentage of instances for which the exact solution was obtained,

10. In previous work (Martins et al., 2011b), we implemented a similar model with a more complex factor graph based on a multi-commodity flow formulation, requiring only the FOL factors described in Section 5.2. In the current paper, we consider a smaller graph with structured factors, which leads to significantly faster runtimes. More involved models, including third-order features, were recently considered in Martins et al. (2013).

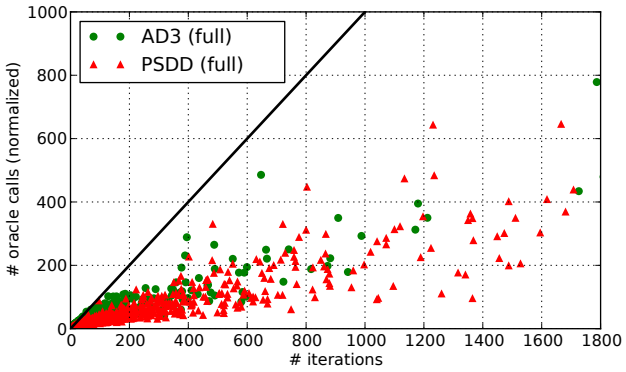


Figure 9: Number of calls to ComputeMAP for AD^3 and PSDD, as a function of the number of iterations. The number of calls is normalized by dividing by the number of factors: in PSDD, this number would equal the number of iterations if there was no caching (black line); each iteration of AD^3 runs 10 iterations of the active set method, thus without caching or warm-starting the normalized number of calls would be ten times the number of AD^3 iterations. Yet, it is clear that both algorithms make significantly fewer calls. Remarkably, after just a few iterations, the number of calls made by the AD^3 and PSDD algorithms are comparable, which means that the number of active set iterations is quickly amortized during the execution of AD^3.

along with a certificate of optimality. For the second-order model, AD^3 was able to solve all the instances to optimality, and in 98.2% of the cases, the LP-MAP was exact. For the full model, AD^3 solved 99.8% of the instances to optimality, being exact in 96.5% of the cases. For the second order model, we obtained in the test set (PTB §23) a parsing speed of 1200 tokens per second and an unlabeled attachment score of 92.48% (fraction of correct dependency attachments excluding punctuation). For the full model, we obtained a speed of 900 tokens per second and a score of 92.62%. All speeds were measured in a desktop PC with Intel Core i7 CPU 3.4 GHz and 8GB RAM. The parser is publicly available as an open-source project at `http://www.ark.cs.cmu.edu/TurboParser`.

8. Discussion and Related Work

We next discuss some of the strengths and weaknesses of AD^3 over other recently proposed LP-MAP inference algorithms. As mentioned in the beginning of Section 4, one of the main sources of difficulty is the *non-smoothness* of the dual objective function (8). This affects both block coordinate descent methods (such as MPLP), which can get stuck at suboptimal stationary points, and the PSDD algorithm, which is tied to the slow $O(1/\epsilon^2)$ convergence of subgradient methods.

Several "smoothing methods" have been proposed in the literature to obviate these drawbacks. Johnson et al. (2007) added an entropic regularization term to the dual objective (8), opening the door for gradient methods; and Jojic et al. (2010) applied an accelerated

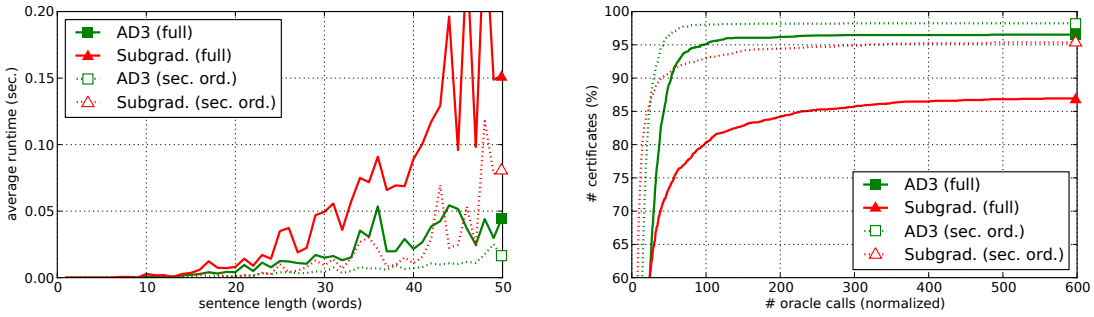


Figure 10: Left: average runtime in PTB §22, as a function of sentence length. Right: percentage of instances, as a function of the normalized number of ComputeMAP calls (see the caption of Figure 9), for which the exact solution was obtained along with a certificate of optimality. The maximum number of iterations is 2000 for both methods.

gradient method to the smoothed problem (Nesterov, 1983), yielding a $O(1/\epsilon)$ iteration bound (the same asymptotic bound as AD^3, as established in Proposition 15). This method has been recently improved by Savchynskyy et al. (2011), through adaptive smoothing and a dynamic estimation of the Lipschitz constant. In a related line of research, Hazan and Shashua (2010) proposed a class of norm-product message-passing algorithms that can be used for both marginal and MAP inference. Norm-product BP implements a primal-dual ascent scheme for optimizing a fractional entropy approximation, constructed as a linear combination of variable and factor entropic terms. For a proper choice of counting numbers, the resulting objective function is convex and smooth, and the amount of smoothness can be controlled by a temperature parameter τ. With $\tau = 0$, norm-product is similar to MPLP and can get stuck at a suboptimal solution; but with a positive τ, the norm-product algorithm is globally convergent to a solution which is $O(\tau)$-close to the LP-MAP optimal value.

Compared with AD^3, the smoothing-based methods mentioned above have the advantage that their local subproblems can typically be transformed into marginal inference problems, which in many cases can be solved with brute-force counting or dynamic programming. However, they also have important drawbacks. First, their precision depends critically on the temperature parameter; e.g., the $O(1/\epsilon)$ iteration bound of Jojic et al. (2010) requires setting the temperature to $O(\epsilon)$, which scales the potentials by $O(1/\epsilon)$ and may lead to numerical instabilities. Second, the solution of the local subproblems are always *dense*; although some marginal values may be low, they are never exactly zero. This contrasts with the projected subgradient and the AD^3 algorithms, for which the solutions of the local subproblems are spanned by one or a small number of MAP configurations. As shown in the experimental section (Figure 9), caching these configurations across iterations may lead to great speedups.

While smoothing-based methods that use quadratic regularizers (as opposed to entropic ones) have also been proposed—most notably the proximal point method of Ravikumar et al.

(2010)—these methods also have disadvantages over AD^3. The proximal point method of Ravikumar et al. (2010) for pairwise MRFs is a double-loop algorithm, where a penalty term with varying magnitude is added to the primal objective, and a globally smooth problem is solved iteratively in the inner loop, using cyclic Bregman projections. Applied to a general factor graph and using a quadratic penalty, the problems solved in the inner loop resemble the AD^3 subproblems, with an important difference: there is an extra Euclidean penalty of the form $\|\boldsymbol{q}_\alpha - \boldsymbol{q}_\alpha^{(t)}\|^2$. While this term makes the subproblems strongly convex, it also destroys the sparsity property mentioned in Proposition 11, which results in substantially more messages needing to be passed around (in particular, messages with size $|\mathcal{Y}_\alpha|$, which can be prohibitive for factors with large degree). A different strategy has been proposed by Pletscher and Wulff (2012), who combined the LP-MAP relaxation described here with a non-convex QP relaxation, which unlike other smoothing methods increases the effect of the penalty term through the progression of the algorithm.

Finally, it should be noted that other strategies have been recently proposed to overcome the weaknesses of coordinate descent algorithms and PSDD, which are not based on smoothing the dual objective. The fact that the PSDD algorithm has "no memory" across iterations (pointed out in the beginning of Section 4) has been addressed by Kappes et al. (2012) in their bundle method, which remembers past updates, at the cost of extra memory storage and more involved local subproblems. The fact that coordinate descent methods can get stuck in suboptimal solutions has been addressed by Schwing et al. (2012), who proposed a ϵ-descent strategy as a way to move away from corners, mixing coordinate and steepest descent steps; the latter, however, require solving QPs as an intermediate step.

During the preparation of this paper, and following our earlier work (Martins et al., 2010, 2011a), AD^3 has been successfully applied to several NLP problems (Martins et al., 2011b, 2013; Das et al., 2012; Almeida and Martins, 2013), and a few related methods have appeared. Meshi and Globerson (2011) also applied ADMM to MAP inference in graphical models, although addressing the dual problem (the one underlying the MPLP algorithm) rather than the primal. Yedidia et al. (2011) proposed the "divide-and-concur" algorithm for LDPC (low-density parity check) decoding, which shares aspects of AD^3, and can be seen as an instance of non-convex ADMM. Barman et al. (2011) proposed an algorithm analogous to AD^3 for the same LDPC decoding problem; their subproblems correspond to projections onto the parity polytope, for which they have derived an efficient algorithm. More recently, Fu et al. (2013) proposed a Bethe-ADMM procedure resembling AD^3, but with an inexact variant of ADMM that makes the subproblems become marginal computations. Recent work also addressed budget and knapsack constraints, important for dealing with cardinality-based potentials and to promote diversity (Tarlow et al., 2010; Almeida and Martins, 2013).

9. Conclusions

We introduced AD^3, a new LP-MAP inference algorithm based on the alternating directions method of multipliers (ADMM) (Glowinski and Marroco, 1975; Gabay and Mercier, 1976).

AD^3 enjoys the modularity of dual decomposition methods, but achieves faster consensus, by penalizing, for each subproblem, deviations from the current global solution. Using recent results, we showed that AD^3 converges to an ϵ-accurate solution with an iteration

bound of $O(1/\epsilon)$. AD^3 can handle factor graphs with hard constraints in first-order logic, using efficient procedures for projecting onto the marginal polytopes of the corresponding factors. This opens the door for using AD^3 in problems with declarative constraints (Roth and Yih, 2004; Richardson and Domingos, 2006). A closed-form solution of the AD^3 subproblem was also derived for pairwise binary factors.

We introduced a new active set method for solving the AD^3 subproblems for arbitrary factors. This method requires only a local MAP oracle, as the PSDD algorithm. The active set method is particularly suitable for these problems, since it can take advantage of warm starting and it deals well with sparse solutions—which are guaranteed by Proposition 11. We also show how AD^3 can be wrapped in a branch-and-bound procedure to retrieve the exact MAP.

Experiments with synthetic and real-world data sets have shown that AD^3 is able to solve the LP-MAP problem more efficiently than other methods for a variety of problems, including MAP inference in Ising and Potts models, protein design, frame-semantic parsing, and dependency parsing.

Our contributions open several directions for future research. One possible extension is to replace the Euclidean penalty of ADMM by a general Mahalanobis distance. The convergence proofs can be trivially extended to Mahalanobis distances, since they correspond to an affine transformation of the subspace defined by the equality constraints of (11). Simple operations, such as scaling these constraints, do not affect the algorithms that are used to solve the subproblems, thus AD^3 can be generalized by including scaling parameters.

Since the AD^3 subproblems can be solved in parallel, significant speed-ups may be obtained in multi-core architectures or using GPU programming. This has been shown to be very useful for large-scale message-passing inference in graphical models (Felzenszwalb and Huttenlocher, 2006; Low et al., 2010; Schwing et al., 2011).

The branch-and-bound algorithm for obtaining the exact MAP deserves further experimental study, as similar approaches have been proven useful in MAP inference problems (Sun et al., 2012). An advantage of AD^3 is its ability to quickly produce sharp upper bounds. For many problems, there are effective rounding procedures that can also produce lower bounds, which can be exploited for guiding the search. There are also alternatives to branch-and-bound, such as tightening procedures (Sontag et al., 2008; Batra et al., 2011), which progressively add larger factors to decrease the duality gap. The variant of AD^3 with the active set method can be used to handle these larger factors.

Acknowledgments

A. M. was supported by the EU/FEDER programme, QREN/POR Lisboa (Portugal), under the Intelligo project (contract 2012/24803) and by FCT grants PTDC/EEI-SII/2312/2012 and UID/EEA/50008/2013. A. M. and M. F. were supported by FCT grant Pest-OE/EEI/0008/2013. N. S. was supported by NSF CAREER IIS-1054319. E. X. was supported by AFOSR FA9550010247, ONR N000140910758, NSF CAREER DBI-0546594, NSF IIS-0713379, and an Alfred P. Sloan Fellowship.

Appendix A. Proof of Convergence Rate of AD^3

In this appendix, we show the $O(1/\epsilon)$ convergence bound of the ADMM algorithm. We use a recent result established by Wang and Banerjee (2012) regarding convergence in a variational setting, from which we derive the convergence of ADMM in the dual objective. We then consider the special case of AD^3, interpreting the constants in the bound in terms of properties of the graphical model.

We start with the following proposition, which states the variational inequality associated with the Lagrangian saddle point problem associated with (10),

$$\min_{\boldsymbol{\lambda}\in\Lambda}\max_{\boldsymbol{q}\in\mathcal{Q},\boldsymbol{p}\in\mathcal{P}} L(\boldsymbol{q},\boldsymbol{p},\boldsymbol{\lambda}), \tag{32}$$

where $L(\boldsymbol{q},\boldsymbol{p},\boldsymbol{\lambda}) := f_1(\boldsymbol{q}) + f_2(\boldsymbol{p}) + \boldsymbol{\lambda}^\top(\mathbf{A}\boldsymbol{q} + \mathbf{B}\boldsymbol{p} - \boldsymbol{c})$ is the standard Lagrangian, and $\Lambda := \{\boldsymbol{\lambda} \mid \max_{\boldsymbol{q}\in\mathcal{Q},\boldsymbol{p}\in\mathcal{P}} L(\boldsymbol{q},\boldsymbol{p},\boldsymbol{\lambda}) < \infty\}$.

Proposition 12 (Variational inequality) *Let $\mathcal{W} := \mathcal{Q}\times\mathcal{P}\times\Lambda$. Given $\boldsymbol{w} = (\boldsymbol{q},\boldsymbol{p},\boldsymbol{\lambda}) \in \mathcal{W}$, define $h(\boldsymbol{w}) := f_1(\boldsymbol{q}) + f_2(\boldsymbol{p})$ and $F(\boldsymbol{w}) := (\mathbf{A}^\top\boldsymbol{\lambda}, \mathbf{B}^\top\boldsymbol{\lambda}, -(\mathbf{A}\boldsymbol{q} + \mathbf{B}\boldsymbol{p} - \boldsymbol{c}))$. Then, $\boldsymbol{w}^* := (\boldsymbol{q}^*,\boldsymbol{p}^*,\boldsymbol{\lambda}^*) \in \mathcal{W}$ is a primal-dual solution of (32) if and only if:*

$$\forall \boldsymbol{w}\in\mathcal{W}, \quad h(\boldsymbol{w}) - h(\boldsymbol{w}^*) + (\boldsymbol{w}-\boldsymbol{w}^*)^\top F(\boldsymbol{w}^*) \le 0. \tag{33}$$

Proof: Assume $\boldsymbol{w}^*$ is a primal-dual solution of (32). Then, the saddle point conditions imply $L(\boldsymbol{q},\boldsymbol{p},\boldsymbol{\lambda}^*) \le L(\boldsymbol{q}^*,\boldsymbol{p}^*,\boldsymbol{\lambda}^*) \le L(\boldsymbol{q}^*,\boldsymbol{p}^*,\boldsymbol{\lambda})$ for every $\boldsymbol{w} := (\boldsymbol{q},\boldsymbol{p},\boldsymbol{\lambda}) \in \mathcal{W}$. Hence:

$$\begin{aligned} 0 &\ge L(\boldsymbol{q},\boldsymbol{p},\boldsymbol{\lambda}^*) - L(\boldsymbol{q}^*,\boldsymbol{p}^*,\boldsymbol{\lambda}) \\ &= f_1(\boldsymbol{q}) + f_2(\boldsymbol{p}) + \boldsymbol{\lambda}^{*\top}(\mathbf{A}\boldsymbol{q} + \mathbf{B}\boldsymbol{p} - \boldsymbol{c}) - f_1(\boldsymbol{q}^*) - f_2(\boldsymbol{p}^*) - \boldsymbol{\lambda}^\top(\mathbf{A}\boldsymbol{q}^* + \mathbf{B}\boldsymbol{p}^* - \boldsymbol{c}) \\ &= h(\boldsymbol{w}) - h(\boldsymbol{w}^*) + (\boldsymbol{w}-\boldsymbol{w}^*)^\top F(\boldsymbol{w}^*). \end{aligned}$$

Conversely, let $\boldsymbol{w}^*$ satisfy (33). Taking $\boldsymbol{w} = (\boldsymbol{q}^*,\boldsymbol{p}^*,\boldsymbol{\lambda})$, we obtain $L(\boldsymbol{q}^*,\boldsymbol{p}^*,\boldsymbol{\lambda}^*) \le L(\boldsymbol{q}^*,\boldsymbol{p}^*,\boldsymbol{\lambda})$. Taking $\boldsymbol{w} = (\boldsymbol{q},\boldsymbol{p},\boldsymbol{\lambda}^*)$, we obtain $L(\boldsymbol{q},\boldsymbol{p},\boldsymbol{\lambda}^*) \le L(\boldsymbol{q}^*,\boldsymbol{p}^*,\boldsymbol{\lambda}^*)$. Hence $(\boldsymbol{q}^*,\boldsymbol{p}^*,\boldsymbol{\lambda}^*)$ is a saddle point, and therefore a primal-dual solution. ■

The next result, due to Wang and Banerjee (2012) and related to previous work by He and Yuan (2011), concerns the convergence rate of ADMM in terms of the variational inequality stated above.

Proposition 13 (Variational convergence rate) *Assume the conditions in Proposition 7. Let $\bar{\boldsymbol{w}}_T = \frac{1}{T}\sum_{t=1}^T \boldsymbol{w}^t$, where $\boldsymbol{w}^t := (\boldsymbol{q}^t,\boldsymbol{p}^t,\boldsymbol{\lambda}^t)$ are the ADMM iterates with $\boldsymbol{\lambda}^0 = \mathbf{0}$. Then, after T iterations:*

$$\forall \boldsymbol{w}\in\mathcal{W}, \quad h(\boldsymbol{w}) - h(\bar{\boldsymbol{w}}_T) + (\boldsymbol{w}-\bar{\boldsymbol{w}}_T)^\top F(\bar{\boldsymbol{w}}_T) \le C/T, \tag{34}$$

where $C = \frac{\eta}{2}\|\mathbf{A}\boldsymbol{q} + \mathbf{B}\boldsymbol{p}^0 - \boldsymbol{c}\|^2 + \frac{1}{2\eta}\|\boldsymbol{\lambda}\|^2$ is independent of T.

Proof: From the variational inequality associated with the $\boldsymbol{q}$-update (13) we have for every $\boldsymbol{q} \in \mathcal{Q}$[11]

$$\begin{aligned}
0 \quad &\geq \quad \nabla_{\boldsymbol{q}} L_\eta(\boldsymbol{q}^{t+1}, \boldsymbol{p}^t, \boldsymbol{\lambda}^t)^\top (\boldsymbol{q} - \boldsymbol{q}^{t+1}) \\
&= \quad \nabla f_1(\boldsymbol{q}^{t+1})^\top (\boldsymbol{q} - \boldsymbol{q}^{t+1}) + (\boldsymbol{q} - \boldsymbol{q}^{t+1})^\top \mathbf{A}^\top (\boldsymbol{\lambda}^t - \eta(\mathbf{A}\boldsymbol{q}^{t+1} + \mathbf{B}\boldsymbol{p}^t - \boldsymbol{c})) \\
&\geq^{(i)} \quad f_1(\boldsymbol{q}) - f_1(\boldsymbol{q}^{t+1}) + (\boldsymbol{q} - \boldsymbol{q}^{t+1})^\top \mathbf{A}^\top (\boldsymbol{\lambda}^t - \eta(\mathbf{A}\boldsymbol{q}^{t+1} + \mathbf{B}\boldsymbol{p}^t - \boldsymbol{c})) \\
&=^{(ii)} \quad f_1(\boldsymbol{q}) - f_1(\boldsymbol{q}^{t+1}) + (\boldsymbol{q} - \boldsymbol{q}^{t+1})^\top \mathbf{A}^\top \boldsymbol{\lambda}^{t+1} - \eta(\mathbf{A}(\boldsymbol{q} - \boldsymbol{q}^{t+1}))^\top \mathbf{B}(\boldsymbol{p}^t - \boldsymbol{p}^{t+1}), \quad (35)
\end{aligned}$$

where in (i) we have used the concavity of f_1, and in (ii) we used (13) for the $\boldsymbol{\lambda}$-updates. Similarly, the variational inequality associated with the $\boldsymbol{p}$-updates (14) yields, for every $\boldsymbol{p} \in \mathcal{P}$:

$$\begin{aligned}
0 \quad &\geq \quad \nabla_{\boldsymbol{p}} L_\eta(\boldsymbol{q}^{t+1}, \boldsymbol{p}^{t+1}, \boldsymbol{\lambda}^t)^\top (\boldsymbol{p} - \boldsymbol{p}^{t+1}) \\
&= \quad \nabla f_2(\boldsymbol{p}^{t+1})^\top (\boldsymbol{p} - \boldsymbol{p}^{t+1}) + (\boldsymbol{p} - \boldsymbol{p}^{t+1})^\top \mathbf{B}^\top (\boldsymbol{\lambda}^t - \eta(\mathbf{A}\boldsymbol{q}^{t+1} + \mathbf{B}\boldsymbol{p}^{t+1} - \boldsymbol{c})) \\
&\geq^{(i)} \quad f_2(\boldsymbol{p}) - f_2(\boldsymbol{p}^{t+1}) + (\boldsymbol{p} - \boldsymbol{p}^{t+1})^\top \mathbf{B}^\top \boldsymbol{\lambda}^{t+1}, \quad (36)
\end{aligned}$$

where in (i) we have used the concavity of f_2. Summing (35) and (36), and noting again that $\boldsymbol{\lambda}^{t+1} = \boldsymbol{\lambda}^t - \eta(\mathbf{A}\boldsymbol{q}^{t+1} + \mathbf{B}\boldsymbol{p}^{t+1} - \boldsymbol{c})$, we obtain, for every $\boldsymbol{w} \in \mathcal{W}$,

$$\begin{aligned}
&h(\boldsymbol{w}^{t+1}) - h(\boldsymbol{w}) + (\boldsymbol{w}^{t+1} - \boldsymbol{w})^\top F(\boldsymbol{w}^{t+1}) \\
&\quad \geq \quad -\eta \mathbf{A}(\boldsymbol{q} - \boldsymbol{q}^{t+1})^\top \mathbf{B}(\boldsymbol{p}^t - \boldsymbol{p}^{t+1}) - \eta^{-1}(\boldsymbol{\lambda} - \boldsymbol{\lambda}^{t+1})^\top (\boldsymbol{\lambda}^{t+1} - \boldsymbol{\lambda}^t). \quad (37)
\end{aligned}$$

We next rewrite the two terms in the right hand side:

$$\begin{aligned}
\eta \mathbf{A}(\boldsymbol{q} - \boldsymbol{q}^{t+1})^\top \mathbf{B}(\boldsymbol{p}^t - \boldsymbol{p}^{t+1}) \quad &= \quad \frac{\eta}{2} \left(\|\mathbf{A}\boldsymbol{q} + \mathbf{B}\boldsymbol{p}^t - \boldsymbol{c}\|^2 - \|\mathbf{A}\boldsymbol{q} + \mathbf{B}\boldsymbol{p}^{t+1} - \boldsymbol{c}\|^2 \right. \\
&+ \quad \left. \|\mathbf{A}\boldsymbol{q}^{t+1} + \mathbf{B}\boldsymbol{p}^{t+1} - \boldsymbol{c}\|^2 - \|\mathbf{A}\boldsymbol{q}^{t+1} + \mathbf{B}\boldsymbol{p}^t - \boldsymbol{c}\|^2 \right); \\
\eta^{-1}(\boldsymbol{\lambda} - \boldsymbol{\lambda}^{t+1})^\top (\boldsymbol{\lambda}^{t+1} - \boldsymbol{\lambda}^t) \quad &= \quad \frac{1}{2\eta} \left(\|\boldsymbol{\lambda} - \boldsymbol{\lambda}^t\|^2 - \|\boldsymbol{\lambda} - \boldsymbol{\lambda}^{t+1}\|^2 - \|\boldsymbol{\lambda}^t - \boldsymbol{\lambda}^{t+1}\|^2 \right).
\end{aligned}$$

Summing (37) over t and noting that $\eta^{-1}\|\boldsymbol{\lambda}^t - \boldsymbol{\lambda}^{t+1}\|^2 = \eta\|\mathbf{A}\boldsymbol{q}^{t+1} + \mathbf{B}\boldsymbol{p}^{t+1} - \boldsymbol{c}\|^2$:

$$\begin{aligned}
&\sum_{t=0}^{T-1} \left(h(\boldsymbol{w}^{t+1}) - h(\boldsymbol{w}) + (\boldsymbol{w}^{t+1} - \boldsymbol{w})^\top F(\boldsymbol{w}^{t+1}) \right) \\
&\quad \geq \quad -\frac{\eta}{2} \left(\|\mathbf{A}\boldsymbol{q} + \mathbf{B}\boldsymbol{p}^0 - \boldsymbol{c}\|^2 - \|\mathbf{A}\boldsymbol{q} + \mathbf{B}\boldsymbol{p}^T - \boldsymbol{c}\|^2 - \sum_{t=0}^{T-1} \|\mathbf{A}\boldsymbol{q}^{t+1} + \mathbf{B}\boldsymbol{p}^t - \boldsymbol{c}\|^2 \right) \\
&\qquad -\frac{1}{2\eta} \left(\|\boldsymbol{\lambda} - \boldsymbol{\lambda}^0\|^2 - \|\boldsymbol{\lambda} - \boldsymbol{\lambda}^T\|^2 \right) \\
&\quad \geq \quad -\frac{\eta}{2} \|\mathbf{A}\boldsymbol{q} + \mathbf{B}\boldsymbol{p}^0 - \boldsymbol{c}\|^2 - \frac{1}{2\eta} \|\boldsymbol{\lambda}\|^2. \quad (38)
\end{aligned}$$

11. Given a problem $\max_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x})$, where f is concave and differentiable and $\mathcal{X}$ is convex, a point $\boldsymbol{x}^* \in \mathcal{X}$ is a maximizer iff it satisfies the variational inequality $\nabla f(\boldsymbol{x}^*)^\top (\boldsymbol{x} - \boldsymbol{x}^*) \leq 0$ for all $\boldsymbol{x} \in \mathcal{X}$ (Facchinei and Pang, 2003).

From the concavity of h, we have that $h(\bar{\boldsymbol{w}}_T) \geq \frac{1}{T}\sum_{t=0}^{T-1} h(\boldsymbol{w}^{t+1})$. Note also that, for every $\tilde{\boldsymbol{w}}$, the function $\boldsymbol{w} \mapsto (\boldsymbol{w}-\tilde{\boldsymbol{w}})^\top F(\boldsymbol{w})$ is affine:

$$\begin{aligned}(\boldsymbol{w}-\tilde{\boldsymbol{w}})^\top F(\boldsymbol{w}) &= (\boldsymbol{q}-\tilde{\boldsymbol{q}})^\top \mathbf{A}^\top\boldsymbol{\lambda} + (\boldsymbol{p}-\tilde{\boldsymbol{p}})^\top \mathbf{B}^\top\boldsymbol{\lambda} - (\boldsymbol{\lambda}-\tilde{\boldsymbol{\lambda}})^\top(\mathbf{A}\boldsymbol{q}+\mathbf{B}\boldsymbol{p}-\boldsymbol{c}) \\ &= -(\mathbf{A}\tilde{\boldsymbol{q}}+\mathbf{B}\tilde{\boldsymbol{p}}-\boldsymbol{c})^\top\boldsymbol{\lambda} + \tilde{\boldsymbol{\lambda}}^\top(\mathbf{A}\boldsymbol{q}+\mathbf{B}\boldsymbol{p}-\boldsymbol{c}) \\ &= F(\tilde{\boldsymbol{w}})^\top\boldsymbol{w} - \boldsymbol{c}^\top\tilde{\boldsymbol{\lambda}}.\end{aligned}$$

As a consequence, $\frac{1}{T}\sum_{t=0}^{T-1}\left(h(\boldsymbol{w}^{t+1}) + (\boldsymbol{w}^{t+1}-\boldsymbol{w})^\top F(\boldsymbol{w}^{t+1})\right) \leq h(\bar{\boldsymbol{w}}_T) + (\bar{\boldsymbol{w}}_T-\boldsymbol{w})^\top F(\bar{\boldsymbol{w}}_T)$, and from (38), we have that $h(\boldsymbol{w}) - h(\bar{\boldsymbol{w}}_T) + (\boldsymbol{w}-\bar{\boldsymbol{w}}_T)^\top F(\bar{\boldsymbol{w}}_T) \leq C/T$, with C as in (34). Note also that, since Λ is convex, we must have $\bar{\boldsymbol{\lambda}}_T \in \Lambda$. ■

Next, we use the bound in Proposition 13 to derive a convergence rate for the dual problem.

Proposition 14 (Dual convergence rate) *Assume the conditions stated in Proposition 13, with $\bar{\boldsymbol{w}}_T$ defined analogously. Let $g : \Lambda \to \mathbb{R}$ be the dual objective function, $g(\boldsymbol{\lambda}) := \max_{\boldsymbol{q}\in\mathcal{Q},\boldsymbol{p}\in\mathcal{P}} L(\boldsymbol{q},\boldsymbol{p},\boldsymbol{\lambda})$, and let $\boldsymbol{\lambda}^* \in \arg\min_{\boldsymbol{\lambda}\in\Lambda} g(\boldsymbol{\lambda})$ be a dual solution. Then, after T iterations, ADMM achieves an $O(\frac{1}{T})$-accurate solution $\bar{\boldsymbol{\lambda}}_T$:*

$$g(\boldsymbol{\lambda}^*) \leq g(\bar{\boldsymbol{\lambda}}_T) \leq g(\boldsymbol{\lambda}^*) + \frac{C}{T},$$

where the constant C is given by

$$C = \frac{5\eta}{2}\left(\max_{\boldsymbol{q}\in\mathcal{Q}} \|\mathbf{A}\boldsymbol{q}+\mathbf{B}\boldsymbol{p}^0-\boldsymbol{c}\|^2\right) + \frac{5}{2\eta}\|\boldsymbol{\lambda}^*\|^2. \tag{39}$$

Proof: By applying Proposition 13 to $\boldsymbol{w} = (\bar{\boldsymbol{q}}_T, \bar{\boldsymbol{p}}_T, \boldsymbol{\lambda})$ we obtain for arbitrary $\boldsymbol{\lambda} \in \Lambda$:

$$-(\boldsymbol{\lambda}-\bar{\boldsymbol{\lambda}}_T)^\top(\mathbf{A}\bar{\boldsymbol{q}}_T+\mathbf{B}\bar{\boldsymbol{p}}_T-\boldsymbol{c}) \leq O(1/T). \tag{40}$$

By applying Proposition 13 to $\boldsymbol{w} = (\boldsymbol{q}, \boldsymbol{p}, \bar{\boldsymbol{\lambda}}_T)$ we obtain for arbitrary $\boldsymbol{q} \in \mathcal{Q}$ and $\boldsymbol{p} \in \mathcal{P}$:

$$\begin{aligned}&f_1(\bar{\boldsymbol{q}}_T) + f_2(\bar{\boldsymbol{p}}_T) + (\mathbf{A}\bar{\boldsymbol{q}}_T+\mathbf{B}\bar{\boldsymbol{p}}_T-\boldsymbol{c})^\top\bar{\boldsymbol{\lambda}}_T \\ &\quad\geq\ f_1(\boldsymbol{q}) + f_2(\boldsymbol{p}) + (\mathbf{A}\boldsymbol{q}+\mathbf{B}\boldsymbol{p}-\boldsymbol{c})^\top\bar{\boldsymbol{\lambda}}_T - O(1/T).\end{aligned}$$

In particular, let $g(\bar{\boldsymbol{\lambda}}_T) = \max_{\boldsymbol{q}\in\mathcal{Q},\boldsymbol{p}\in\mathcal{P}} L(\boldsymbol{q},\boldsymbol{p},\bar{\boldsymbol{\lambda}}_T) = L(\widehat{\boldsymbol{q}}_T, \widehat{\boldsymbol{p}}_T, \bar{\boldsymbol{\lambda}}_T)$ be the value of the dual objective at $\bar{\boldsymbol{\lambda}}_T$, where $(\widehat{\boldsymbol{q}}_T, \widehat{\boldsymbol{p}}_T)$ are the corresponding maximizers. We then have:

$$f_1(\bar{\boldsymbol{q}}_T) + f_2(\bar{\boldsymbol{p}}_T) + (\mathbf{A}\bar{\boldsymbol{q}}_T+\mathbf{B}\bar{\boldsymbol{p}}_T-\boldsymbol{c})^\top\bar{\boldsymbol{\lambda}}_T \geq g(\bar{\boldsymbol{\lambda}}_T) - O(1/T). \tag{41}$$

Finally we have (letting $\boldsymbol{w}^* = (\boldsymbol{q}^*, \boldsymbol{p}^*, \boldsymbol{\lambda}^*)$ be the optimal primal-dual solution):

$$\begin{aligned}g(\boldsymbol{\lambda}^*) &= \max_{\boldsymbol{q}\in\mathcal{Q},\boldsymbol{p}\in\mathcal{P}} f_1(\boldsymbol{q}) + f_2(\boldsymbol{p}) + \boldsymbol{\lambda}^{*\top}(\mathbf{A}\boldsymbol{q}+\mathbf{B}\boldsymbol{p}-\boldsymbol{c}) \\ &\geq f_1(\bar{\boldsymbol{q}}_T) + f_2(\bar{\boldsymbol{p}}_T) + \boldsymbol{\lambda}^{*\top}(\mathbf{A}\bar{\boldsymbol{q}}_T+\mathbf{B}\bar{\boldsymbol{p}}_T-\boldsymbol{c}) \\ &\geq^{(i)} f_1(\bar{\boldsymbol{q}}_T) + f_2(\bar{\boldsymbol{p}}_T) + \bar{\boldsymbol{\lambda}}_T^\top(\mathbf{A}\bar{\boldsymbol{q}}_T+\mathbf{B}\bar{\boldsymbol{p}}_T-\boldsymbol{c}) - O(1/T) \\ &\geq^{(ii)} g(\bar{\boldsymbol{\lambda}}_T) - O(1/T),\end{aligned}$$

where in (i) we used (40) and in (ii) we used (41). By definition of $\boldsymbol{\lambda}^*$, we also have $g(\bar{\boldsymbol{\lambda}}_T) \geq g(\boldsymbol{\lambda}^*)$. Since we applied Proposition 13 twice, the constant inside the O-notation becomes

$$C = \frac{\eta}{2}\left(\|\mathbf{A}\bar{\boldsymbol{q}}_T + \mathbf{B}\boldsymbol{p}^0 - \boldsymbol{c}\|^2 + \|\mathbf{A}\widehat{\boldsymbol{q}}_T + \mathbf{B}\boldsymbol{p}^0 - \boldsymbol{c}\|^2\right) + \frac{1}{2\eta}\left(\|\boldsymbol{\lambda}^*\|^2 + \|\bar{\boldsymbol{\lambda}}_T\|^2\right). \tag{42}$$

Even though C depends on $\bar{\boldsymbol{q}}_T$, $\widehat{\boldsymbol{q}}_T$, and $\bar{\boldsymbol{\lambda}}_T$, it is easy to obtain an upper bound on C when $\mathcal{Q}$ is a bounded set, using the fact that the sequence $(\boldsymbol{\lambda}^t)_{t\in\mathbb{N}}$ is bounded by a constant, which implies that the average $\bar{\boldsymbol{\lambda}}_T$ is also bounded. Indeed, from Boyd et al. (2011, p.107), we have that

$$V^t := \eta^{-1}\|\boldsymbol{\lambda}^* - \boldsymbol{\lambda}^t\|^2 + \eta\|\mathbf{B}(\boldsymbol{p}^* - \boldsymbol{p}^t)\|^2$$

is a Lyapunov function, i.e., $0 \leq V^{t+1} \leq V^t$ for every $t \in \mathbb{N}$. This implies that $V^t \leq V^0 = \eta^{-1}\|\boldsymbol{\lambda}^*\|^2 + \eta\|\mathbf{B}(\boldsymbol{p}^* - \boldsymbol{p}^0)\|^2$; since $V^t \geq \eta^{-1}\|\boldsymbol{\lambda}^* - \boldsymbol{\lambda}^t\|^2$, we can replace above and write:

$$\begin{aligned} 0 &\geq \|\boldsymbol{\lambda}^* - \boldsymbol{\lambda}^t\|^2 - \|\boldsymbol{\lambda}^*\|^2 - \eta^2\|\mathbf{B}(\boldsymbol{p}^* - \boldsymbol{p}^0)\|^2 = \|\boldsymbol{\lambda}^t\|^2 - 2\boldsymbol{\lambda}^{*\top}\boldsymbol{\lambda}^t - \eta^2\|\mathbf{B}(\boldsymbol{p}^* - \boldsymbol{p}^0)\|^2 \\ &\geq \|\boldsymbol{\lambda}^t\|^2 - 2\|\boldsymbol{\lambda}^*\|\|\boldsymbol{\lambda}^t\| - \eta^2\|\mathbf{B}(\boldsymbol{p}^* - \boldsymbol{p}^0)\|^2, \end{aligned}$$

where in the last line we invoked the Cauchy-Schwarz inequality. Solving the quadratic equation, we obtain $\|\boldsymbol{\lambda}^t\| \leq \|\boldsymbol{\lambda}^*\| + \sqrt{\|\boldsymbol{\lambda}^*\|^2 + \eta^2\|\mathbf{B}(\boldsymbol{p}^0 - \boldsymbol{p}^*)\|^2}$, which in turn implies

$$\begin{aligned} \|\boldsymbol{\lambda}^t\|^2 &\leq 2\|\boldsymbol{\lambda}^*\|^2 + \eta^2\|\mathbf{B}(\boldsymbol{p}^0 - \boldsymbol{p}^*)\|^2 + 2\|\boldsymbol{\lambda}^*\|\sqrt{\|\boldsymbol{\lambda}^*\|^2 + \eta^2\|\mathbf{B}(\boldsymbol{p}^0 - \boldsymbol{p}^*)\|^2} \\ &\leq 2\|\boldsymbol{\lambda}^*\|^2 + \eta^2\|\mathbf{B}(\boldsymbol{p}^0 - \boldsymbol{p}^*)\|^2 + 2(\|\boldsymbol{\lambda}^*\|^2 + \eta^2\|\mathbf{B}(\boldsymbol{p}^0 - \boldsymbol{p}^*)\|^2) \\ &= 4\|\boldsymbol{\lambda}^*\|^2 + 3\eta^2\|\mathbf{A}\boldsymbol{q}^* + \mathbf{B}\boldsymbol{p}^0 - \boldsymbol{c}\|^2, \end{aligned} \tag{43}$$

the last line following from $\mathbf{A}\boldsymbol{q}^* + \mathbf{B}\boldsymbol{p}^* = \boldsymbol{c}$. Replacing (43) in (42) yields the result. ■

Finally, we will see how the bounds above apply to the AD^3 algorithm.

Proposition 15 (Dual convergence rate of AD^3) *After T iterations of AD^3, we achieve an $O(\frac{1}{T})$-accurate solution $\bar{\boldsymbol{\lambda}}_T := \sum_{t=0}^{T-1}\boldsymbol{\lambda}^{(t)}$:*

$$g(\boldsymbol{\lambda}^*) \leq g(\bar{\boldsymbol{\lambda}}_T) \leq g(\boldsymbol{\lambda}^*) + \frac{C}{T},$$

where $C = \frac{5\eta}{2}\sum_i |\partial(i)|(1 - |\mathcal{Y}_i|^{-1}) + \frac{5}{2\eta}\|\boldsymbol{\lambda}^*\|^2$ is a constant independent of T.

Proof: With the uniform initialization of the $\boldsymbol{p}$-variables in AD^3, the first term of (39) is maximized by a choice of $\boldsymbol{q}_\alpha$-variables that puts all mass in a single configuration. That is:

$$\max_{\boldsymbol{q}_{i\alpha}} \|\boldsymbol{q}_{i\alpha} - |\mathcal{Y}_i|^{-1}\mathbf{1}\|^2 = \left((1 - |\mathcal{Y}_i|^{-1})^2 + (|\mathcal{Y}_i| - 1)|\mathcal{Y}_i|^{-2}\right) = 1 - |\mathcal{Y}_i|^{-1}.$$

This leads to the desired bound. ■

Appendix B. Derivation of Solutions for AD^3 Subproblems

B.1 Binary Pairwise Factors

In this section, we prove Proposition 10. Let us first assume that $c_{12} \geq 0$. In this case, the constraints $z_{12} \geq z_1 + z_2 - 1$ and $z_{12} \geq 0$ in (21) are always inactive and the problem can be simplified to:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}(z_1 - c_1)^2 + \frac{1}{2}(z_2 - c_2)^2 - c_{12} z_{12} \\ \text{with respect to} \quad & z_1, z_2, z_{12} \\ \text{subject to} \quad & z_{12} \leq z_1, \quad z_{12} \leq z_2, \quad z_1 \in [0,1], \quad z_2 \in [0,1]. \end{aligned} \tag{44}$$

If $c_{12} = 0$, the problem becomes separable, and a solution is

$$z_1^* = [c_1]_{\mathbb{U}}, \quad z_2^* = [c_2]_{\mathbb{U}}, \quad z_{12}^* = \min\{z_1^*, z_2^*\},$$

which complies with (22). We next analyze the case where $c_{12} > 0$. The Lagrangian of (44) is:

$$\begin{aligned} L(\boldsymbol{z}, \boldsymbol{\mu}, \boldsymbol{\lambda}, \boldsymbol{\nu}) \quad = \quad & \frac{1}{2}(z_1 - c_1)^2 + \frac{1}{2}(z_2 - c_2)^2 - c_{12} z_{12} + \mu_1(z_{12} - z_1) + \mu_2(z_{12} - z_2) \\ & - \lambda_1 z_1 - \lambda_2 z_2 + \nu_1(z_1 - 1) + \nu_2(z_2 - 1). \end{aligned}$$

At optimality, the following Karush-Kuhn-Tucker (KKT) conditions need to be satisfied:

$$\nabla_{z_1} L(\boldsymbol{z}^*, \boldsymbol{\mu}^*, \boldsymbol{\lambda}^*, \boldsymbol{\nu}^*) = 0 \quad \Rightarrow \quad z_1^* = c_1 + \mu_1^* + \lambda_1^* - \nu_1^* \tag{45}$$

$$\nabla_{z_2} L(\boldsymbol{z}^*, \boldsymbol{\mu}^*, \boldsymbol{\lambda}^*, \boldsymbol{\nu}^*) = 0 \quad \Rightarrow \quad z_2^* = c_2 + \mu_2^* + \lambda_2^* - \nu_2^* \tag{46}$$

$$\nabla_{z_{12}} L(\boldsymbol{z}^*, \boldsymbol{\mu}^*, \boldsymbol{\lambda}^*, \boldsymbol{\nu}^*) = 0 \quad \Rightarrow \quad c_{12} = \mu_1^* + \mu_2^* \tag{47}$$

$$\lambda_1^* z_1^* = 0, \quad \lambda_2^* z_2^* = 0 \tag{48}$$

$$\mu_1^*(z_{12}^* - z_1^*) = 0, \quad \mu_2^*(z_{12}^* - z_2^*) = 0 \tag{49}$$

$$\nu_1^*(z_1^* - 1) = 0, \quad \nu_2^*(z_2^* - 1) = 0 \tag{50}$$

$$\boldsymbol{\mu}^*, \boldsymbol{\lambda}^*, \boldsymbol{\nu}^* \geq 0 \tag{51}$$

$$z_{12}^* \leq z_1^*, \quad z_{12}^* \leq z_2^*, \quad z_1^* \in [0,1], \quad z_2^* \in [0,1] \tag{52}$$

We are going to consider three cases separately:

1. $\boxed{z_1^* > z_2^*}$ From the primal feasibility conditions (52), this implies $z_1^* > 0$, $z_2^* < 1$, and $z_{12}^* < z_1^*$. Complementary slackness (48,49,50) implies in turn $\lambda_1^* = 0$, $\nu_2^* = 0$, and $\mu_1^* = 0$. From (47) we have $\mu_2^* = c_{12}$. Since we are assuming $c_{12} > 0$, we then have $\mu_2^* > 0$, and complementary slackness (49) implies $z_{12}^* = z_2^*$. Plugging this into (45)–(46) we obtain

$$z_1^* = c_1 - \nu_1^* \leq c_1, \qquad z_2^* = c_2 + \lambda_2^* + c_{12} \geq c_2 + c_{12}.$$

Now we have the following:

- Either $z_1^* = 1$ or $z_1^* < 1$. In the latter case, $\nu_1^* = 0$ due to (50), hence $z_1^* = c_1$. Since in any case we must have $z_1^* \leq c_1$, we conclude that $z_1^* = \min\{c_1, 1\}$.

- Either $z_2^* = 0$ or $z_2^* > 0$. In the latter case, $\lambda_2^* = 0$ due to (48), hence $z_2^* = c_2 + c_{12}$. Since in any case we must have $z_2^* \geq \lambda_2$, we conclude that $z_2^* = \max\{0, c_2 + c_{12}\}$.

In sum:

$$z_1^* = \min\{c_1, 1\}, \quad z_{12}^* = z_2^* = \max\{0, c_2 + c_{12}\},$$

and our assumption $z_1^* > z_2^*$ can only be valid if $c_1 > c_2 + c_{12}$.

2. $\boxed{z_1^* < z_2^*}$ By symmetry, we have

$$z_2^* = \min\{c_2, 1\}, \quad z_{12}^* = z_1^* = \max\{0, c_1 + c_{12}\},$$

and our assumption $z_1^* < z_2^*$ can only be valid if $c_2 > c_1 + c_{12}$.

3. $\boxed{z_1^* = z_2^*}$ In this case, it is easy to verify that we must have $z_{12}^* = z_1^* = z_2^*$, and we can rewrite our optimization problem in terms of one variable only (call it z). The problem becomes that of minimizing $\frac{1}{2}(z - c_1)^2 + \frac{1}{2}(z - c_2)^2 - c_{12}z$, which equals a constant plus $(z - \frac{c_1 + c_2 + c_{12}}{2})^2$, subject to $z \in \mathbb{U} \triangleq [0, 1]$. Hence:

$$z_{12}^* = z_1^* = z_2^* = [(c_1 + c_2 + c_{12})/2]_{\mathbb{U}}\,.$$

Putting all the pieces together, we obtain the solution displayed in (22).

It remains to address the case where $c_{12} < 0$. By redefining $c_1' = c_1 + c_{12}$, $c_2' = 1 - c_2$, $c_{12}' = -c_{12}$, $z_2' = 1 - z_2$, and $z_{12}' = z_1 - z_{12}$, we can reduce (21) to the form in (44). Substituting back in (22), we obtain the solution displayed in (23).

B.2 Marginal Polytope of Hard Constraint Factors

The following proposition establishes that the marginal polytope of a hard constraint factor is the convex hull of its acceptance set.

Proposition 16 *Let α be a binary hard constraint factor with degree K, and consider the set of all possible distributions $\mathbb{P}(\boldsymbol{Y}_\alpha)$ which satisfy $\mathbb{P}(\boldsymbol{Y}_\alpha = \boldsymbol{y}_\alpha) = 0$ for every $\boldsymbol{y}_\alpha \notin \mathcal{S}_\alpha$. Then, the set of possible marginals realizable for some distribution in that set is given by*

$$\begin{aligned} \mathcal{Z}_\alpha &:= \left\{ (q_{1\alpha}(1), \ldots, q_{K\alpha}(1)) \;\middle|\; \boldsymbol{q}_{i\alpha} = \mathbf{M}_{i\alpha}\boldsymbol{q}_\alpha, \text{ for some } \boldsymbol{q}_\alpha \in \Delta^{|\mathcal{Y}_\alpha|} \text{ s.t. } q_\alpha(\boldsymbol{y}_\alpha) = 0, \forall \boldsymbol{y}_\alpha \notin \mathcal{S}_\alpha \right\} \\ &= \operatorname{conv} \mathcal{S}_\alpha. \end{aligned}$$

Proof: From the fact that we are constraining $q_\alpha(\boldsymbol{y}_\alpha) = 0, \forall \boldsymbol{y}_\alpha \notin \mathcal{S}_\alpha$, it follows:

$$\begin{aligned} \mathcal{Z}_\alpha &= \left\{ \boldsymbol{z} \geq 0 \;\middle|\; \exists \boldsymbol{q}_\alpha \geq 0 \text{ s.t. } \forall i \in \partial(\alpha), z_i = \sum_{\substack{\boldsymbol{y}_\alpha \in \mathcal{S}_\alpha \\ \boldsymbol{y}_i = 1}} q_\alpha(\boldsymbol{y}_\alpha) = 1 - \sum_{\substack{\boldsymbol{y}_\alpha \in \mathcal{S}_\alpha \\ \boldsymbol{y}_i = 0}} q_\alpha(\boldsymbol{y}_\alpha) \right\} \\ &= \left\{ \boldsymbol{z} \geq 0 \;\middle|\; \exists \boldsymbol{q}_\alpha \geq 0, \sum_{\boldsymbol{y}_\alpha \in \mathcal{S}_\alpha} q_\alpha(\boldsymbol{y}_\alpha) = 1 \text{ s.t. } \boldsymbol{z} = \sum_{\boldsymbol{y}_\alpha \in \mathcal{S}_\alpha} q_\alpha(\boldsymbol{y}_\alpha)\boldsymbol{y}_\alpha \right\} \\ &= \operatorname{conv} \mathcal{S}_\alpha. \end{aligned}$$

■

Algorithm 4 Projection onto simplex (Duchi et al., 2008)

Input: $\boldsymbol{z}_0$
Sort $\boldsymbol{z}_0$ into $\boldsymbol{y}_0$: $y_1 \geq \ldots \geq y_K$
Find $\rho = \max\left\{j \in [K] \mid y_{0j} - \frac{1}{j}\left(\left(\sum_{r=1}^{j} y_{0r}\right) - 1\right) > 0\right\}$
Define $\tau = \frac{1}{\rho}\left(\sum_{r=1}^{\rho} y_{0r} - 1\right)$
Output: $\boldsymbol{z}$ defined as $z_i = \max\{z_{0i} - \tau, 0\}$.

For hard constraint factors, the AD^3 subproblems take the following form (cf. (20)):

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}\sum_{i\in\partial(\alpha)} \|\boldsymbol{q}_{i\alpha} - \boldsymbol{a}_i\|^2 \quad \text{with respect to} \quad \boldsymbol{q}_\alpha \in \Delta^{|\mathcal{Y}_\alpha|},\ \boldsymbol{q}_{i\alpha} \in \mathbb{R}^{|\mathcal{Y}_i|},\ \forall i \in \partial(\alpha) \\ &\text{subject to} \quad \boldsymbol{q}_{i\alpha} = \mathbf{M}_{i\alpha}\boldsymbol{q}_\alpha, \quad q_\alpha(\boldsymbol{y}_\alpha) = 0,\ \forall \boldsymbol{y}_\alpha \neq \mathcal{S}_\alpha. \end{aligned}$$

From Proposition 16, and making use of a reduced parametrization, noting that $\|\boldsymbol{q}_{i\alpha} - \boldsymbol{a}_i\|^2 = (q_{i\alpha}(1) - a_i(1))^2 + (1 - q_{i\alpha}(1) - a_i(0))^2$, which equals a constant plus $2(q_{i\alpha}(1) - (a_i(1) + 1 - a_i(0))/2)^2$, we have that this problem is equivalent to:

$$\text{minimize} \quad \frac{1}{2}\|\boldsymbol{z} - \boldsymbol{z}_0\|^2 \quad \text{with respect to} \quad \boldsymbol{z} \in \mathcal{Z}_\alpha,$$

where $z_{0i} := (a_i(1) + 1 - a_i(0))/2$, for each $i \in \partial(\alpha)$.

B.3 XOR Factor

For the XOR factor, the quadratic problem in (20) reduces to that of *projecting onto the simplex*. That problem is well-known in the optimization community (see, e.g., Brucker 1984; Michelot 1986); by writing the KKT conditions, it is simple to show that the solution $\boldsymbol{z}^*$ is a soft-thresholding of $\boldsymbol{z}_0$, and therefore the problem can be reduced to that of finding the right threshold. Algorithm 4 provides an efficient procedure; it requires a sort operation, which renders its cost $O(K \log K)$. A proof of correctness appears in Duchi et al. (2008).[12]

B.4 OR Factor

The following procedure can be used for computing a projection onto $\mathcal{Z}_{\text{OR}}$:

1. Set $\tilde{\boldsymbol{z}}$ as the projection of $\boldsymbol{z}_0$ onto the unit cube. This can be done by clipping each coordinate to the unit interval $\mathbb{U} = [0, 1]$, i.e., by setting $\tilde{z}_i = [z_{0i}]_{\mathbb{U}} = \min\{1, \max\{0, z_{0i}\}\}$. If $\sum_{i=1}^{K} \tilde{z}_i \geq 1$, then return $\tilde{\boldsymbol{z}}$. Else go to step 2.
2. Return the projection of $\boldsymbol{z}_0$ onto the simplex (use Algorithm 4).

The correctness of this procedure is justified by the following lemma:

Lemma 17 (Sifting Lemma.) *Consider a problem of the form*

$$P: \qquad \min_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x}) \quad \textit{subject to} \quad g(\boldsymbol{x}) \leq 0, \tag{53}$$

12. This cost can be reduced to $O(K)$ using linear-time selection algorithms (Pardalos and Kovoor, 1990).

where $\mathcal{X}$ is nonempty convex subset of $\mathbb{R}^D$ and $f : \mathcal{X} \to \mathbb{R}$ and $g : \mathcal{X} \to \mathbb{R}$ are convex functions. Suppose that the problem (53) *is feasible and bounded below, and let $\mathcal{A}$ be the set of solutions of the* relaxed problem $\min_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x})$*, i.e. $\mathcal{A} = \{\boldsymbol{x} \in \mathcal{X} \mid f(\boldsymbol{x}) \leq f(\boldsymbol{x}'),\ \forall \boldsymbol{x}' \in \mathcal{X}\}$. Then:*

1. *if for some $\tilde{\boldsymbol{x}} \in \mathcal{A}$ we have $g(\tilde{\boldsymbol{x}}) \leq 0$, then $\tilde{\boldsymbol{x}}$ is also a solution of the original problem P;*

2. *otherwise (if for all $\tilde{\boldsymbol{x}} \in \mathcal{A}$ we have $g(\tilde{\boldsymbol{x}}) > 0$), then the inequality constraint is necessarily active in P, i.e., problem P is equivalent to* $\min_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x})$ *subject to $g(\boldsymbol{x}) = 0$.*

Proof: Let f^* be the optimal value of P. The first statement is obvious: since $\tilde{\boldsymbol{x}}$ is a solution of a relaxed problem we have $f(\tilde{\boldsymbol{x}}) \leq f^*$; hence if $\tilde{\boldsymbol{x}}$ is feasible this becomes an equality. For the second statement, assume that $\exists \boldsymbol{x} \in \mathcal{X}$ subject to $g(\boldsymbol{x}) < 0$ (otherwise, the statement holds trivially). The nonlinear Farkas' lemma (Bertsekas et al., 2003, Prop. 3.5.4, p. 204) implies that there exists some $\lambda^* \geq 0$ subject to $f(\boldsymbol{x}) - f^* + \lambda^* g(\boldsymbol{x}) \geq 0$ holds for all $\boldsymbol{x} \in \mathcal{X}$. In particular, this also holds for an optimal $\boldsymbol{x}^*$ (i.e., such that $f^* = f(\boldsymbol{x}^*)$), which implies that $\lambda^* g(\boldsymbol{x}^*) \geq 0$. However, since $\lambda^* \geq 0$ and $g(\boldsymbol{x}^*) \leq 0$ (since $\boldsymbol{x}^*$ has to be feasible), we also have $\lambda^* g(\boldsymbol{x}^*) \leq 0$, i.e., $\lambda^* g(\boldsymbol{x}^*) = 0$. Now suppose that $\lambda^* = 0$. Then we have $f(\boldsymbol{x}) - f^* \geq 0$, $\forall \boldsymbol{x} \in \mathcal{X}$, which implies that $\boldsymbol{x}^* \in \mathcal{A}$ and contradicts the assumption that $g(\tilde{\boldsymbol{x}}) > 0, \forall \tilde{\boldsymbol{x}} \in \mathcal{A}$. Hence we must have $g(\boldsymbol{x}^*) = 0$. ■

Let us see how the Sifting Lemma applies to the problem of projecting onto $\mathcal{Z}_{\text{OR}}$. If the relaxed problem in the first step does not return a feasible point then, from the Sifting Lemma, the constraint $\sum_{i=1}^K z_i \geq 1$ has to be active, i.e., we must have $\sum_{i=1}^K z_i = 1$. This, in turn, implies that $\boldsymbol{z} \leq 1$, hence the problem becomes equivalent to the XOR case. In sum, the worst-case runtime is $O(K \log K)$, although it is $O(K)$ if the first step succeeds.

B.5 OR-with-output Factor

Solving the AD^3 subproblem for the OR-with-output factor is slightly more complicated than in the previous cases; however, we next see that it can also be addressed in $O(K \log K)$ time with a sort operation. The polytope $\mathcal{Z}_{\text{OR}-\text{out}}$ can be expressed as the intersection of the following three sets:[13]

$$\begin{aligned}
\mathbb{U}^{K+1} &:= [0,1]^{K+1} \\
\mathcal{A}_1 &:= \{\boldsymbol{z} \in \mathbb{R}^{K+1} \mid z_k \leq z_{K+1}, \forall k = 1, \ldots, K\} \\
\mathcal{A}_2 &:= \left\{\boldsymbol{z} \in [0,1]^{K+1} \;\middle|\; \textstyle\sum_{k=1}^K z_k \geq z_{K+1}\right\}.
\end{aligned}$$

We further define $\mathcal{A}_0 := [0,1]^{K+1} \cap \mathcal{A}_1$, and we denote by $\text{proj}_{\mathcal{Z}}(\boldsymbol{z})$ the Euclidean projection of a point $\boldsymbol{z}$ onto a convex set $\mathcal{Z}$. From Lemma 17, we have that the following procedure is correct:

13. Actually, the set $\mathbb{U}^{K+1}$ is redundant, since we have $\mathcal{A}_2 \subseteq \mathbb{U}^{K+1}$ and therefore $\mathcal{Z}_{\text{OR}-\text{out}} = \mathcal{A}_1 \cap \mathcal{A}_2$. However it is computationally advantageous to consider this redundancy, as we shall see.

1. Set $\tilde{\boldsymbol{z}} := \mathrm{proj}_{\mathbb{U}^{K+1}}(\boldsymbol{z}_0)$. If $\tilde{\boldsymbol{z}} \in \mathcal{A}_1 \cap \mathcal{A}_2$, then we are done: just return $\tilde{\boldsymbol{z}}$. Else, if $\tilde{\boldsymbol{z}} \in \mathcal{A}_1$ but $\tilde{\boldsymbol{z}} \notin \mathcal{A}_2$, discard $\tilde{\boldsymbol{z}}$ and go to step 3. Otherwise, discard $\tilde{\boldsymbol{z}}$ and go to step 2.

2. Set $\tilde{\boldsymbol{z}} := \mathrm{proj}_{\mathcal{A}_0}(\boldsymbol{z}_0)$ (we will describe later how to compute this projection). If $\tilde{\boldsymbol{z}} \in \mathcal{A}_2$, return $\tilde{\boldsymbol{z}}$. Otherwise, discard $\tilde{\boldsymbol{z}}$ and go to step 3.

3. Set $\tilde{\boldsymbol{z}} := \mathrm{proj}_{\bar{\mathcal{A}}_2}(\boldsymbol{z}_0)$, where $\bar{\mathcal{A}}_2 := \{\boldsymbol{z} \in [0,1]^{K+1} \mid \sum_{k=1}^{K} z_k = z_{K+1}\}$ (this set is precisely the marginal polytope of a XOR factor with the last output negated, hence the projection corresponds to the local subproblem for that factor, for which we can employ Algorithm 4).

Note that the first step above can be omitted; however, it avoids performing step 2 (which requires a sort) unless it is really necessary. To completely specify the algorithm, we only need to explain how to compute the projection onto $\mathcal{A}_0$ (step 2). The next proposition states that this can be done by first projecting onto $\mathcal{A}_1$, and then projecting the result onto $[0,1]^{K+1}$.

We first start with a lemma establishing a sufficient condition for the composition of two individual projections be equivalent to projecting onto the intersection (which is not true in general).[14]

Lemma 18 *Let $X \subseteq \mathbb{R}^D$ and $Y \subseteq \mathbb{R}^D$ be convex sets, and suppose $\boldsymbol{z}^* = \mathrm{proj}_Y(\boldsymbol{z}_0 + \boldsymbol{z}^* - \boldsymbol{z}')$ holds for any $\boldsymbol{z}_0 \in \mathbb{R}^D$, where $\boldsymbol{z}' = \mathrm{proj}_Y(\boldsymbol{z}_0)$, and $\boldsymbol{z}^* = \mathrm{proj}_X(\boldsymbol{z}')$. Then, we have $\mathrm{proj}_{X \cap Y} = \mathrm{proj}_X \circ \mathrm{proj}_Y$.*

Proof: Assume $\boldsymbol{z}^* = \mathrm{proj}_Y(\boldsymbol{z}_0 + \boldsymbol{z}^* - \boldsymbol{z}')$. Then, we have $(\boldsymbol{z}_0 + \boldsymbol{z}^* - \boldsymbol{z}' - \boldsymbol{z}^*)^\top (\boldsymbol{z} - \boldsymbol{z}^*) \leq 0$ for all $\boldsymbol{z} \in Y$; in particular, $(\boldsymbol{z}_0 - \boldsymbol{z}')^\top (\boldsymbol{z} - \boldsymbol{z}^*) \leq 0$ for all $\boldsymbol{z} \in X \cap Y$. On the other hand, the definition of $\boldsymbol{z}^*$ implies $(\boldsymbol{z}' - \boldsymbol{z}^*)^\top (\boldsymbol{z} - \boldsymbol{z}^*) \leq 0$ for all $\boldsymbol{z} \in X$, and in particular for $\boldsymbol{z} \in X \cap Y$. Summing these two inequalities, we obtain $(\boldsymbol{z}_0 - \boldsymbol{z}^*)^\top (\boldsymbol{z} - \boldsymbol{z}^*) \leq 0$ for all $\boldsymbol{z} \in X \cap Y$, that is, $\boldsymbol{z}^* = \mathrm{proj}_{X \cap Y}(\boldsymbol{z}_0)$. ■

Proposition 19 *It holds $\mathrm{proj}_{\mathcal{A}_0} = \mathrm{proj}_{\mathbb{U}^{K+1}} \circ \mathrm{proj}_{\mathcal{A}_1}$. Furthermore, a projection onto $\mathcal{A}_1$ can be computed in $O(K \log K)$ time using Algorithm 5.*

Proof: We first prove the second part. Note that a projection onto $\mathcal{A}_1$ can be written as the following problem:

$$\text{minimize} \quad \frac{1}{2}\|\boldsymbol{z} - \boldsymbol{z}_0\|^2 \quad \text{subject to} \quad z_k \leq z_{K+1}, \ \forall k = 1, \ldots, K, \tag{54}$$

14. This is equivalent to Dykstra's projection algorithm (Boyle and Dykstra, 1986) converging in one iteration.

and we have successively:

$$\begin{aligned}
\min_{z_k \le z_{K+1},\, \forall k} \frac{1}{2}\|\boldsymbol{z}-\boldsymbol{z}_0\|^2 &= \min_{z_{K+1}} \frac{1}{2}(z_{K+1}-z_{0,K+1})^2 + \sum_{k=1}^{K} \min_{z_k \le z_{K+1}} \frac{1}{2}(z_k - z_{0k})^2 \\
&= \min_{z_{K+1}} \frac{1}{2}(z_{K+1}-z_{0,K+1})^2 + \sum_{k=1}^{K} \frac{1}{2}(\min\{z_{K+1}, z_{0k}\} - z_{0k})^2 \\
&= \min_{z_{K+1}} \frac{1}{2}(z_{K+1}-z_{0,K+1})^2 + \frac{1}{2}\sum_{k \in \mathcal{I}(z_{K+1})} (z_{K+1}-z_{0k})^2.
\end{aligned}$$

where $\mathcal{I}(z_{K+1}) \triangleq \{k \in [K] : z_{0k} \ge z_{K+1}\}$. Assuming that the set $\mathcal{I}(z_{K+1})$ is given, the previous is a sum-of-squares problem whose solution is

$$z^*_{K+1} = \frac{z_{0,K+1} + \sum_{k \in \mathcal{I}(z_{K+1})} z_{0k}}{1 + |\mathcal{I}(z_{K+1})|}.$$

The set $\mathcal{I}(z_{K+1})$ can be determined by inspection after sorting $z_{01}, \ldots, z_{0K}$. The procedure is shown in Algorithm 5.

To prove the first part, we invoke Lemma 18. It suffices to show that $\boldsymbol{z}^* = \mathrm{proj}_{\mathcal{A}_1}(\boldsymbol{z}_0 + \boldsymbol{z}^* - \boldsymbol{z}')$ holds for any $\boldsymbol{z}_0 \in \mathbb{R}^D$, where $\boldsymbol{z}' = \mathrm{proj}_{\mathcal{A}_1}(\boldsymbol{z}_0)$, and $\boldsymbol{z}^* = \mathrm{proj}_{\mathbb{U}^{K+1}}(\boldsymbol{z}')$. Looking at Algorithm 5, we see that:

$$z'_k = \begin{cases} \tau, & \text{if } k = K+1 \text{ or } z_{0k} \ge \tau \\ z_{0k}, & \text{otherwise,} \end{cases} \qquad z^*_k = [z'_k]_{\mathbb{U}} = \begin{cases} [\tau]_{\mathbb{U}}, & \text{if } k = K+1 \text{ or } z_{0k} \ge \tau \\ [z_{0k}]_{\mathbb{U}}\,, & \text{otherwise.} \end{cases}$$

$$z_{0k} + z^*_k - z'_k = \begin{cases} [\tau]_{\mathbb{U}} - \tau + z_{0k}, & \text{if } k = K+1 \text{ or } z_{0k} \ge \tau \\ [z_{0k}]_{\mathbb{U}}\,, & \text{otherwise.} \end{cases}$$

Now two things should be noted about Algorithm 5:

- If a constant is added to all entries in $\boldsymbol{z}_0$, the set $\mathcal{I}(z_{K+1})$ remains the same, and τ and $\boldsymbol{z}$ are affected by the same constant;
- Let $\tilde{\boldsymbol{z}}_0$ be such that $\tilde{z}_{0k} = z_{0k}$ if $k = K+1$ or $z_{0k} \ge \tau$, and $\tilde{z}_{0k} \le \tau$ otherwise. Let $\tilde{\boldsymbol{z}}$ be the projected point when such $\tilde{\boldsymbol{z}}_0$ is given as input. Then $\mathcal{I}(\tilde{z}_{K+1}) = \mathcal{I}(z_{K+1})$, $\tilde{\tau} = \tau$, $\tilde{z}_k = z_k$ if $k = K+1$ or $z_{0k} \ge \tau$, and $\tilde{z}_k = \tilde{z}_{0k}$ otherwise.

The two facts above allow to relate the projection of $\boldsymbol{z}_0 + \boldsymbol{z}^* - \boldsymbol{z}'$ with that of $\boldsymbol{z}_0$. Using $[\tau]_{\mathbb{U}} - \tau$ as the constant, and noting that, for $k \neq K+1$ and $z_{0k} < \tau$, we have $[z_{0k}]_{\mathbb{U}} - [\tau]_{\mathbb{U}} + \tau \ge \tau$ if $z_{0k} < \tau$, the two facts imply that:

$$\mathrm{proj}_{\mathcal{A}_1}(\boldsymbol{z}_0 + \boldsymbol{z}^* - \boldsymbol{z}') = \begin{cases} z'_k + [\tau]_{\mathbb{U}} - \tau = [\tau]_{\mathbb{U}}, & \text{if } k = K+1 \text{ or } z_{0k} \ge \tau \\ [z_{0k}]_{\mathbb{U}}\,, & \text{otherwise;} \end{cases}$$

hence $\boldsymbol{z}^* = \mathrm{proj}_{\mathcal{A}_1}(\boldsymbol{z}_0 + \boldsymbol{z}^* - \boldsymbol{z}')$, which concludes the proof. ■

Algorithm 5 Projection onto $\mathcal{A}_1$

Input: $\boldsymbol{z}_0$
Sort $z_{01}, \ldots, z_{0K}$ into $y_1 \geq \ldots \geq y_K$
Find $\rho = \min \left\{ j \in [K+1] \mid \frac{1}{j} \left(z_{0,K+1} + \sum_{r=1}^{j-1} y_r \right) > y_j \right\}$
Define $\tau = \frac{1}{\rho} \left(z_{0,K+1} + \sum_{r=1}^{\rho-1} y_r \right)$
Output: $\boldsymbol{z}$ defined as $z_{K+1} = \tau$ and $z_i = \min\{z_{0i}, \tau\}$, $i = 1, \ldots, K$.

Appendix C. Proof of Proposition 11

We first show that the rank of the matrix $\mathbf{M}$ is at most $\sum_{i \in \partial(\alpha)} |\mathcal{Y}_i| - \partial(\alpha) + 1$. For each $i \in \partial(\alpha)$, let us consider the $|\mathcal{Y}_i|$ rows of $\mathbf{M}$. By definition of $\mathbf{M}$, the set of entries on these rows which have the value 1 form a partition of $\mathcal{Y}_\alpha$, hence, summing these rows yields the all-ones row vector, and this happens for each $i \in \partial(\alpha)$. Hence we have at least $\partial(\alpha) - 1$ rows that are linearly dependent. This shows that the rank of $\mathbf{M}$ is at most $\sum_{i \in \partial(\alpha)} |\mathcal{Y}_i| - \partial(\alpha) + 1$. Let us now rewrite (20) as

$$\text{minimize} \quad \frac{1}{2}\|\boldsymbol{u} - \boldsymbol{a}\|^2 + g(\boldsymbol{u}) \quad \text{with respect to} \quad \boldsymbol{u} \in \mathbb{R}^{\sum_i |\mathcal{Y}_i|}, \tag{55}$$

where $g(\boldsymbol{u})$ is the solution value of the following linear problem:

$$\begin{aligned} &\text{minimize} \quad -\boldsymbol{b}^\top \boldsymbol{q}_\alpha \quad \text{with respect to} \quad \boldsymbol{q}_\alpha \in \mathbb{R}^{|\mathcal{Y}_\alpha|} \\ &\text{subject to} \quad \begin{cases} \mathbf{M}\boldsymbol{q}_{\boldsymbol{\alpha}} = \boldsymbol{u} \\ \mathbf{1}^\top \boldsymbol{q}_{\boldsymbol{\alpha}} = 1 \\ \boldsymbol{q}_{\boldsymbol{\alpha}} \geq 0. \end{cases} \end{aligned} \tag{56}$$

From the simplex constraints (last two lines), we have that problem (56) is bounded below (i.e., $g(\boldsymbol{u}) > -\infty$). Furthermore, problem (56) is feasible (i.e., $g(\boldsymbol{u}) < +\infty$) iff $\boldsymbol{u} \in \prod_{i \in \partial(\alpha)} \Delta^{|\mathcal{Y}_i|}$, which in turn implies $\mathbf{1}^\top \boldsymbol{q}_\alpha = 1$. Hence we can add these constraints to the problem in (55), discard the constraint $\mathbf{1}^\top \boldsymbol{q}_\alpha = 1$ in (56), and assume that the resulting problem (which we reproduce below) is feasible and bounded below:

$$\begin{aligned} &\text{minimize} \quad -\boldsymbol{b}^\top \boldsymbol{q}_\alpha \quad \text{with respect to} \quad \boldsymbol{q}_\alpha \in \mathbb{R}^{|\mathcal{Y}_\alpha|} \\ &\text{subject to} \quad \mathbf{M}\boldsymbol{q}_\alpha = \boldsymbol{u}, \quad \boldsymbol{q}_\alpha \geq 0. \end{aligned} \tag{57}$$

Problem (57) is a linear program in standard form. Since it is feasible and bounded, it admits a solution at a vertex of the constraint set (Rockafellar, 1970). We have that a feasible point $\widehat{\boldsymbol{q}}_\alpha$ is a vertex if and only if the columns of $\mathbf{M}$ indexed by $\{\boldsymbol{y}_\alpha \mid \widehat{\boldsymbol{q}}\alpha(\boldsymbol{y}_\alpha) \neq 0\}$ are linearly independent. We cannot have more than $\sum_{i \in \partial(\alpha)} |\mathcal{Y}_i| - \partial(\alpha) + 1$ of these columns, since this is the rank of $\mathbf{M}$. It follows that (57) (and hence (20)) has a solution $\boldsymbol{q}^*_\alpha$ with at most $\sum_{i \in \partial(\alpha)} |\mathcal{Y}_i| - \partial(\alpha) + 1$ nonzeros.

References

M. B. Almeida and A. F. T. Martins. Fast and robust compressive summarization with dual decomposition and multi-task learning. In *Proc. of the Annual Meeting of the Association for Computational Linguistics*, 2013.

S. Barman, X. Liu, S. Draper, and B. Recht. Decomposition methods for large scale LP decoding. In *49th Annual Allerton Conference on Communication, Control, and Computing*, pages 253–260. IEEE, 2011.

D. Batra, S. Nowozin, and P. Kohli. Tighter relaxations for MAP-MRF inference: A local primal-dual gap based separation algorithm. In *International Conference on Artificial Intelligence and Statistics*, pages 146–154, 2011.

D. Bertsekas, W. Hager, and O. Mangasarian. *Nonlinear Programming*. Athena Scientific, 1999.

D.P. Bertsekas, A. Nedic, and A.E. Ozdaglar. *Convex Analysis and Optimization*. Athena Scientific, 2003.

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*. Now Publishers, 2011.

S. P. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

J.P. Boyle and R.L. Dykstra. A method for finding projections onto the intersections of convex sets in Hilbert spaces. In *Advances in Order Restricted Statistical Inference*, pages 28–47. Springer Verlag, 1986.

P. Brucker. An $O(n)$ algorithm for quadratic knapsack problems. *Operations Research Letters*, 3(3):163–166, 1984.

M. Chang, L. Ratinov, and D. Roth. Constraints as prior knowledge. In *International Conference of Machine Learning: Workshop on Prior Knowledge for Text and Language Processing*, July 2008.

Y. J. Chu and T. H. Liu. On the shortest arborescence of a directed graph. *Science Sinica*, 14:1396–1400, 1965.

D. Das. *Semi-Supervised and Latent-Variable Models of Natural Language Semantics*. PhD thesis, Carnegie Mellon University, 2012.

D. Das, A.F.T. Martins, and N.A. Smith. An exact dual decomposition algorithm for shallow semantic parsing with constraints. In *Proc. of First Joint Conference on Lexical and Computational Semantics (*SEM)*, 2012.

J. Duchi, D. Tarlow, G. Elidan, and D. Koller. Using combinatorial optimization within max-product belief propagation. *Advances in Neural Information Processing Systems*, 19, 2007.

J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the L1-ball for learning in high dimensions. In *Proc. of International Conference of Machine Learning*, 2008.

J. Eckstein and D. Bertsekas. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(1): 293–318, 1992.

J. Edmonds. Optimum branchings. *Journal of Research of the National Bureau of Standards*, 71B:233–240, 1967.

J.M. Eisner. Three new probabilistic models for dependency parsing: An exploration. In *Proc. of International Conference on Computational Linguistics*, pages 340–345, 1996.

F. Facchinei and J.S. Pang. *Finite-Dimensional Variational Inequalities and Complementarity Problems*, volume 1. Springer Verlag, 2003.

P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. *International Journal of Computer Vision*, 70(1):41–54, 2006.

C.J. Fillmore. Frame semantics and the nature of language. *Annals of the New York Academy of Sciences*, 280(1):20–32, 1976.

Q. Fu, H. Wang, and A. Banerjee. Bethe-ADMM for tree decomposition based parallel MAP inference. In *Proc. of Uncertainty in Artificial Intelligence*, 2013.

D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers and Mathematics with Applications*, 2(1): 17–40, 1976.

A. Globerson and T. Jaakkola. Fixing max-product: Convergent message passing algorithms for MAP LP-relaxations. *Neural Information Processing Systems*, 20, 2008.

R. Glowinski and A. Marroco. Sur l'approximation, par éléments finis d'ordre un, et la résolution, par penalisation-dualité, d'une classe de problèmes de Dirichlet non linéaires. *Rev. Franc. Automat. Inform. Rech. Operat.*, 9:41–76, 1975.

T. Hazan and A. Shashua. Norm-product belief propagation: Primal-dual message-passing for approximate inference. *IEEE Transactions on Information Theory*, 56(12):6294–6316, 2010.

B.S. He and X.M. Yuan. On the $O(1/t)$ convergence rate of alternating direction method. *SIAM Journal of Numerical Analysis (to appear)*, 2011.

M. Hestenes. Multiplier and gradient methods. *Journal of Optimization Theory and Applications*, 4:302–320, 1969.

J.K. Johnson, D.M. Malioutov, and A.S. Willsky. Lagrangian relaxation for MAP estimation in graphical models. In *45th Annual Allerton Conference on Communication, Control and Computing*, 2007.

V. Jojic, S. Gould, and D. Koller. Accelerated dual decomposition for MAP inference. In *International Conference of Machine Learning*, 2010.

J. Kappes, B. Savchynskyy, and C. Schnorr. A bundle approach to efficient MAP-inference by Lagrangian relaxation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009.

V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:1568–1583, 2006.

N. Komodakis, N. Paragios, and G. Tziritas. MRF optimization via dual decomposition: Message-passing revisited. In *Proc. of International Conference on Computer Vision*, 2007.

T. Koo, A. M. Rush, M. Collins, T. Jaakkola, and D. Sontag. Dual decomposition for parsing with non-projective head automata. In *Proc. of Empirical Methods for Natural Language Processing*, 2010.

V.A. Kovalevsky and V.K. Koval. A diffusion algorithm for decreasing energy of max-sum labeling problem. Technical report, Glushkov Institute of Cybernetics, Kiev, USSR, 1975.

F. R. Kschischang, B. J. Frey, and H. A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47, 2001.

A. Kulesza and F. Pereira. Structured learning with approximate inference. *Neural Information Processing Systems*, 2007.

S. Lauritzen. *Graphical Models*. Clarendon Press, Oxford, 1996. ISBN 0-19-852219-3.

Y. Low, J. Gonzalez, A. Kyrola, D. Bickson, C. Guestrin, and J.M. Hellerstein. Graphlab: A new parallel framework for machine learning. In *International Conference on Uncertainty in Artificial Intelligence*, 2010.

M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini. Building a large annotated corpus of English: the Penn treebank. *Computational Linguistics*, 19(2):313–330, 1993.

A. F. T. Martins. *The Geometry of Constrained Structured Prediction: Applications to Inference and Learning of Natural Language Syntax*. PhD thesis, Carnegie Mellon University and Instituto Superior Técnico, 2012.

A. F. T. Martins, N. A. Smith, and E. P. Xing. Polyhedral outer approximations with application to natural language parsing. In *Proc. of International Conference of Machine Learning*, 2009.

A. F. T. Martins, N. A. Smith, E. P. Xing, P. M. Q. Aguiar, and M. A. T. Figueiredo. Augmented dual decomposition for MAP inference. In *Neural Information Processing Systems: Workshop in Optimization for Machine Learning*, 2010.

A. F. T. Martins, M. A. T. Figueiredo, P. M. Q. Aguiar, N. A. Smith, and E. P. Xing. An augmented Lagrangian approach to constrained MAP inference. In *Proc. of International Conference on Machine Learning*, 2011a.

A. F. T. Martins, N. A. Smith, P. M. Q. Aguiar, and M. A. T. Figueiredo. Dual decomposition with many overlapping components. In *Proc. of Empirical Methods for Natural Language Processing*, 2011b.

A. F. T. Martins, M. B. Almeida, and N. A. Smith. Turning on the turbo: Fast third-order non-projective turbo parsers. In *Proc. of the Annual Meeting of the Association for Computational Linguistics*, 2013.

R. McDonald and G. Satta. On the complexity of non-projective data-driven dependency parsing. In *Proc. of International Conference on Parsing Technologies*, 2007.

R. T. McDonald, F. Pereira, K. Ribarov, and J. Hajic. Non-projective dependency parsing using spanning tree algorithms. In *Proc. of Empirical Methods for Natural Language Processing*, 2005.

O. Meshi and A. Globerson. An alternating direction method for dual MAP LP relaxation. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, 2011.

C. Michelot. A finite algorithm for finding the projection of a point onto the canonical simplex of $\mathbb{R}^n$. *Journal of Optimization Theory and Applications*, 50(1):195–200, 1986.

Y. Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Math. Doklady*, 27:372–376, 1983.

J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer verlag, 1999.

S. Nowozin and C.H. Lampert. Global connectivity potentials for random field models. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 818–825. IEEE, 2009.

P. M. Pardalos and N. Kovoor. An algorithm for a singly constrained class of quadratic programs subject to upper and lower bounds. *Mathematical Programming*, 46(1):321–328, 1990.

J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

P. Pletscher and S. Wulff. LPQP for MAP: Putting LP Solvers to Better Use. In *Proc. of International Conference on Machine Learning*, 2012.

H. Poon and P. Domingos. Unsupervised semantic parsing. In *Proc. of Empirical Methods in Natural Language Processing*, 2009.

M. Powell. A method for nonlinear constraints in minimization problems. In R. Fletcher, editor, *Optimization*, pages 283–298. Academic Press, 1969.

V. Punyakanok, D. Roth, W. Yih, and D. Zimak. Learning and inference over constrained output. In *Proc. of International Joint Conference on Artificial Intelligence*, 2005.

P. Ravikumar, A. Agarwal, and M. Wainwright. Message-passing for graph-structured linear programs: Proximal methods and rounding schemes. *Journal of Machine Learning Research*, 11:1043–1080, 2010.

M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 62(1):107–136, 2006.

T.J. Richardson and R.L. Urbanke. *Modern Coding Theory*. Cambridge University Press, 2008.

R.T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

D. Roth and W. Yih. A linear programming formulation for global inference in natural language tasks. In *International Conference on Natural Language Learning*, 2004.

A.M. Rush and M. Collins. A tutorial on dual decomposition and Lagrangian relaxation for inference in natural language processing. *Journal of Artificial Intelligence Research*, 45:305–362, 2012.

B. Savchynskyy, S. Schmidt, J. Kappes, and C. Schnörr. A study of Nesterov's scheme for Lagrangian decomposition and MAP labeling. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

M. Schlesinger. Syntactic analysis of two-dimensional visual signals in noisy conditions. *Kibernetika*, 4:113–130, 1976.

A. Schwing, T. Hazan, M. Pollefeys, and R. Urtasun. Distributed message passing for large scale graphical models. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1833–1840, 2011.

A. Schwing, T. Hazan, M. Pollefeys, and R. Urtasun. Globally convergent dual MAP LP relaxation solvers using Fenchel-Young margins. In *Advances in Neural Information Processing Systems 25*, pages 2393–2401, 2012.

D. Smith and J. Eisner. Dependency parsing by belief propagation. In *Proc. of Empirical Methods for Natural Language Processing*, 2008.

D. Sontag, T. Meltzer, A. Globerson, Y. Weiss, and T Jaakkola. Tightening LP relaxations for MAP using message-passing. In *Proc. of Uncertainty in Artificial Intelligence*, 2008.

D. Sontag, A. Globerson, and T. Jaakkola. Introduction to dual decomposition for inference. In *Optimization for Machine Learning*. MIT Press, 2011.

M. Sun, M. Telaprolu, H. Lee, and S. Savarese. An efficient branch-and-bound algorithm for optimal human pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1616–1623, 2012.

R. Tanner. A recursive approach to low complexity codes. *IEEE Transactions on Information Theory*, 27(5):533–547, 1981.

D. Tarlow, I. E. Givoni, and R. S. Zemel. HOP-MAP: Efficient message passing with high order potentials. In *AISTATS*, 2010.

M. Wainwright and M. Jordan. *Graphical Models, Exponential Families, and Variational Inference.* Now Publishers, 2008.

M. Wainwright, T. Jaakkola, and A. Willsky. MAP estimation via agreement on trees: message-passing and linear programming. *IEEE Transactions on Information Theory*, 51 (11):3697–3717, 2005.

H. Wang and A. Banerjee. Online alternating direction method. In *Proc. of International Conference on Machine Learning*, 2012.

T. Werner. A linear programming approach to max-sum problem: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:1165–1179, 2007.

H. Yamada and Y. Matsumoto. Statistical dependency analysis with support vector machines. In *Proc. of International Conference on Parsing Technologies*, 2003.

C. Yanover, T. Meltzer, and Y. Weiss. Linear programming relaxations and belief propagation—an empirical study. *Journal of Machine Learning Research*, 7:1887–1907, 2006.

J.S. Yedidia, Y. Wang, and S.C. Draper. Divide and concur and difference-map BP decoders for LDPC codes. *IEEE Transactions on Information Theory*, 57(2):786–802, 2011.

Journal of Machine Learning Research 16 (2015) 547-551 Submitted 7/13; Published 3/15

Introducing CURRENNT: The Munich Open-Source CUDA RecurREnt Neural Network Toolkit

Felix Weninger WENINGER@TUM.DE
Johannes Bergmann PRIVAT@JOHANNES-BERGMANN.DE
Björn Schuller* SCHULLER@TUM.DE
Machine Learning & Signal Processing, Technische Universität München, 80290 Munich, Germany

Editor: Mikio Braun

Abstract

In this article, we introduce CURRENNT, an open-source parallel implementation of deep recurrent neural networks (RNNs) supporting graphics processing units (GPUs) through NVIDIA's Computed Unified Device Architecture (CUDA). CURRENNT supports uni- and bidirectional RNNs with Long Short-Term Memory (LSTM) memory cells which overcome the vanishing gradient problem. To our knowledge, CURRENNT is the first publicly available parallel implementation of deep LSTM-RNNs. Benchmarks are given on a noisy speech recognition task from the 2013 2nd CHiME Speech Separation and Recognition Challenge, where LSTM-RNNs have been shown to deliver best performance. In the result, double digit speedups in bidirectional LSTM training are achieved with respect to a reference single-threaded CPU implementation. CURRENNT is available under the GNU General Public License from http://sourceforge.net/p/currennt.

Keywords: parallel computing, deep neural networks, recurrent neural networks, Long Short-Term Memory

1. Introduction

Recurrent neural networks (RNNs) are known as powerful sequence learners. In particular, the Long Short-Term Memory (LSTM) architecture has been proven to provide excellent modeling of language (Sundermeyer et al., 2012), music (Eck and Schmidhuber, 2002), speech (Graves et al., 2013), and facial expressions (Wöllmer et al., 2012). LSTM units overcome the vanishing gradient problem of traditional RNNs by the introduction of a memory cell which can be controlled by input, output and reset operations (Gers et al., 2000). In particular, recent research demonstrates that deep LSTM-RNNs exhibit superior performance in speech recognition in comparison to state-of-the-art deep feed forward networks (Graves et al., 2013). However, in contrast to the widespread usage of the latter (Hinton et al., 2012), RNNs are still not adopted by the research community at large. One of the major barriers is the lack of high-performance implementations for training RNNs; at the same time, such implementations are non-trivial due to the limited parallelism caused by time dependencies. To the best of our knowledge, there is no publicly available software dedicated to parallel LSTM-RNN training. Thus, we introduce our CUDA RecurREnt Neural Network Toolkit (CURRENNT) which exploits a mini-batch learning scheme performing parallel weight

∗. B. Schuller is also with the Department of Computing, Imperial College London, UK.

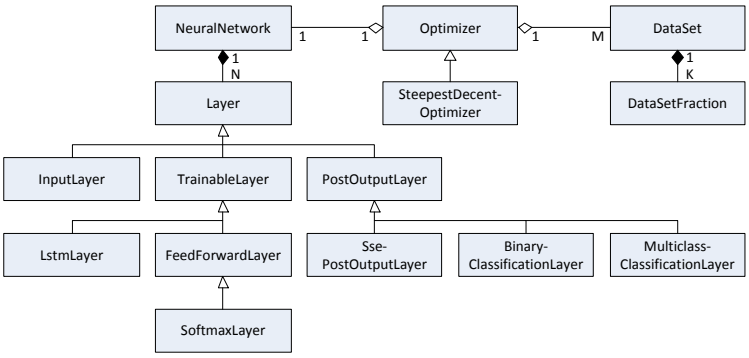


Figure 1: CURRENNT's C++ classes for deep feedforward and LSTM-RNN modeling.

updates from multiple sequences. CURRENNT implements learning from big data sets that do not fit into memory by means of a random access data format. GPUs are supported through NVIDIA's Computed Unified Device Architecture (CUDA). The RNN structure implemented in CURRENNT is based on LSTM units and in addition, feedforward network training is supported. Besides simple regression, CURRENNT also includes logistic and softmax output layers for training of binary and multi-way classification.

To briefly refer some related studies and freely available implementations: A 'reference' CPU implementation of LSTM-RNNs as used by Graves (2008) is available as open-source C++ code (Graves, 2013). A Python library for many machine learning algorithms including LSTM-RNN has been introduced by Schaul et al. (2010); however, it does not directly support parallel processing. Multi-core training of (standard) RNNs has been investigated by Cernanský (2009), but the source code is not available. Pascanu et al. have recently released a Python implementation ('GroundHog') of various RNN types described in their study (Pascanu et al., 2014), exploiting GPU-accelerated training through Theano; yet, it does not provide LSTM-RNNs, and at the moment there is no user-friendly interface.

2. Design

CURRENNT provides a C++ class library for deep LSTM-RNN modeling (cf. Figure 1) and a command line application for network training and evaluation. The network architecture can be specified by the user in JavaScript Object Notation (JSON), and trained parameters are saved in the same format, allowing, e.g., for deep learning with pre-training. For efficiency reasons, features for training and evaluation are given in binary format, adhering to the NetCDF standard, but network outputs can also be saved in CSV format to facilitate post-processing. All C++ code is designed to be platform independent and has been tested on Windows and various Linux distributions. The required CUDA compute capability is 1.3 (2008), allowing usage on virtually all of the consumer grade NVIDIA GPUs deployed in today's desktop PCs. The behavior of the gradient descent training algorithm is controlled by various switches of the command line application, allowing, e.g., for on-line or batch learning and fine-tuning of the training algorithm such as adding Gaussian noise to the input activations and randomly shuffling training data in on-line learning to improve generalization (Graves et al., 2013). The interested reader is referred to the documentation for more details.

3. Implementation

Deep LSTM-RNNs with N layers are implemented as follows. An input sequence $\mathbf{x}_t$ is mapped to the output sequence $\mathbf{y}_t$, $t = 1, \ldots, T$ through the iteration (*forward pass*):

$$\begin{aligned} \mathbf{h}_t^{(0)} &:= \mathbf{x}_t, \\ \mathbf{h}_t^{(n)} &:= \mathcal{L}_t^{(n)} \left(\mathbf{h}_t^{(n-1)}, \mathbf{h}_{t-1}^{(n)} \right), \\ \mathbf{y}_t &:= \mathcal{S} \left(\mathbf{W}^{(N),(N+1)} \mathbf{h}_t^{(N)} + \mathbf{b}^{(N+1)} \right). \end{aligned}$$

In the above and the ongoing, $\mathbf{W}$ denotes weight matrices and $\mathbf{b}$ stands for bias vectors (with superscripts denoting layer indices). $\mathbf{h}_t^{(n)}$ denotes the hidden feature representation of time frame t in the level n units, $n = 1, \ldots, N$. The 0-th layer is the input layer and the $N+1$-th layer the output layer. $\mathcal{S}$ is the (vector valued) output layer function, e.g., a softmax function for multi-way classification (cf. Figure 1). $\mathcal{L}_t^{(n)}$ denotes the composite LSTM activation function which is used instead of the common simple sigmoid-shaped functions. The crucial point is to augment each unit with a state variable c_t, resulting in an automaton-like structure. The hidden layer activations correspond to the state variables ('memory cells') scaled by the activations of the 'output gates' $\mathbf{o}_t^{(n)}$,

$$\begin{aligned} \mathbf{h}_t^{(n)} &= \mathbf{o}_t^{(n)} \otimes \tanh(\mathbf{c}_t^{(n)}), \\ \mathbf{c}_t^{(n)} &= \mathbf{f}_t^{(n)} \otimes \mathbf{c}_{t-1}^{(n)} + \mathbf{i}_t^{(n)} \otimes \tanh \left(\mathbf{W}^{(n-1),(n)} \mathbf{h}_t^{(n-1)} + \mathbf{W}^{(n),(n)} \mathbf{h}_{t-1}^{(n)} + \mathbf{b}_c^{(n)} \right), \end{aligned} \quad (1)$$

where $\otimes$ denotes element-wise multiplication and tanh is also applied element-wise. Thus, the state is scaled by a 'forget' gate (Gers et al., 2000) with dynamic activation $\mathbf{f}_t^{(n)}$ instead of a recurrent connection with static weight. $\mathbf{i}_t^{(n)}$ is the activation of the input gate that regulates the 'influx' from the feedforward and recurrent connections. The activations of the input, output and forget gates are calculated in a similar fashion as $\mathbf{c}_t^{(n)}$ (Graves et al., 2013). From the dependencies between layers ($n-1 \rightsquigarrow n$) and time steps ($t-1 \rightsquigarrow t$) in the above, it is obvious that parallel computation of feedforward activations cannot be performed across layers; further, parallel computation of recurrent activations is not possible across time steps. Thus, to increase the degree of parallelization, we consider *data fractions* (cf. Figure 1) of size P out of S sequences in parallel, each having exactly T time steps (creating 'dummy' time steps for shorter sequences which are neglected in the error calculation). For instance, we consider a state matrix $\mathbf{C}^{(n)}$ for the n-th layer,

$$\mathbf{C}^{(n)} = [\mathbf{c}_{1,p}^{(n)} \cdots \mathbf{c}_{1,p+P-1}^{(n)} \cdots \mathbf{c}_{T,p}^{(n)} \cdots \mathbf{c}_{T,p+P-1}^{(n)}], \quad (2)$$

where $\mathbf{c}_{t,p}^{(n)}$ is the state for sequence p in layer n at time t. To realize the update equation (1) we can now compute the feedforward part for all time steps and P sequences in parallel simply by pre-multiplication with $\mathbf{W}^{(n-1),(n)}$. For the recurrent part, we can update $\mathbf{C}^{(n)}$ from 'left to right' using $\mathbf{W}^{(n),(n)}$. Input, output and forget gate activations are calculated in an analogous fashion. In this process, the matrix structure (2) ensures memory locality of the data corresponding to one time step (matrices are stored in column-major order). For bidirectional layers the above matrix structure is replicated at each layer; in the 'backward'

	RNNLIB (Graves, 2013)	CURRENNT			
Parallel sequences (P)	1	1	10	50	200
Validation set error (10 ep.)	0.138	0.138	0.135	0.137	0.144
Validation set error (50 ep.)	0.120	0.119	0.116	0.118	0.119
Training time / epoch [s]	7 420	3 805	580	392	334
Speedup	(1.0)	2.0	12.8	18.9	22.2

Table 1: Performance (error / speedup) on CHiME 2013 noisy speech recognition task.

part, the recurrent parts are updated from 'right to left', and activations are collected in a single vector before passing them to the subsequent layer (Graves et al., 2013).

During network training, the *backward pass* for the hidden layers is realized similarly, by splitting the matrix of weight changes into a part propagated to the preceding layer and a recurrent part propagated to the previous time step, resulting in a parallel implementation of the backpropagation through time (BPTT) algorithm. The weight changes are applied for all sequences (batch learning) or for each data fraction. Thus, if $1 < P < S$ we perform mini-batch learning. In this case, only P sequences have to be kept in memory at once, allowing for learning from large data sets.

4. Benchmark

We conclude with a benchmark on a word recognition task with convolutive non-stationary noise from the 2013 2nd CHiME Challenge's track 1 (Vincent et al., 2013), where bidirectional LSTM decoding has been shown to deliver best performance (Geiger et al., 2013). We consider frame-wise word error rate as well as computation speedup in training with respect to the open source C++ reference implementation by Graves (2013) running in a single CPU thread on an Intel Core2Quad PC with 4 GB of RAM. The GPU is an NVIDIA GTX 560 with 2 GB of RAM. We compare results for different values of P while fixing the other training parameters. The corresponding NetCDF, network configuration, and training parameter files are distributed with CURRENNT. Results (Figure 1) show that the error rate after 50 epochs is not heavily influenced by the batch size for parallel processing, while speedups of up to 22.2 can be achieved.

5. Conclusions

CURRENNT, our GPU implementation of deep LSTM-RNN for labeling sequential data, has been shown to deliver double digit training speedups at equal accuracy in a noisy speech recognition task. Future work will be concentrated on discriminative training objectives and cost functions for transcription tasks (Graves, 2008).

Acknowledgments

This research has been supported by DFG grant SCHU 2502/4-1. The authors would like to thank Alex Graves for helpful discussions.

References

M. Cernanský. Training recurrent neural network using multistream extended Kalman filter on multicore processor and CUDA enabled graphic processor unit. In *Proc. of ICANN*, volume 1, pages 381–390, 2009.

D. Eck and J. Schmidhuber. Learning the long-term structure of the blues. In *Proc. of ICANN*, pages 284–289, 2002.

J. T. Geiger, F. Weninger, A. Hurmalainen, J. F. Gemmeke, M. Wöllmer, B. Schuller, G. Rigoll, and T. Virtanen. The TUM+TUT+KUL approach to the CHiME Challenge 2013: Multi-stream ASR exploiting BLSTM networks and sparse NMF. In *Proc. 2nd CHiME Workshop*, pages 25–30, Vancouver, Canada, 2013.

F. Gers, J. Schmidhuber, and F. Cummins. Learning to forget: Continual prediction with LSTM. *Neural Computation*, 12(10):2451–2471, 2000.

A. Graves. *Supervised Sequence Labelling with Recurrent Neural Networks*. PhD thesis, Technische Universität München, 2008.

A. Graves. RNNLIB: A recurrent neural network library for sequence learning problems. `http://sourceforge.net/projects/rnnl/`, 2013.

A. Graves, A. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *Proc. of ICASSP*, pages 6645–6649, Vancouver, Canada, 2013.

G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio. How to construct deep recurrent neural networks. In *Proc. of ICLR*, 2014.

T. Schaul, J. Bayer, D. Wierstra, Y. Sun, M. Felder, F. Sehnke, T. Rückstieß, and J. Schmidhuber. PyBrain. *Journal of Machine Learning Research*, 11:743–746, 2010.

M. Sundermeyer, R. Schlüter, and H. Ney. LSTM neural networks for language modeling. In *Proc. of INTERSPEECH*, Portland, OR, USA, 2012.

E. Vincent, J. Barker, S. Watanabe, J. Le Roux, F. Nesta, and M. Matassoni. The second 'CHiME' speech separation and recognition challenge: Datasets, tasks and baselines. In *Proc. of ICASSP*, pages 126–130, Vancouver, Canada, 2013.

M. Wöllmer, M. Kaiser, F. Eyben, F. Weninger, B. Schuller, and G. Rigoll. Fully automatic audiovisual emotion recognition – voice, words, and the face. In T. Fingscheidt and W. Kellermann, editors, *Proceedings of Speech Communication; 10. ITG Symposium*, pages 1–4, Braunschweig, Germany, 2012.

Journal of Machine Learning Research 16 (2015) 553-557 Submitted 11/12; Revised 3/14; Published 3/15

The `flare` Package for High Dimensional Linear Regression and Precision Matrix Estimation in `R`*

Xingguo Li† LIXX1661@UMN.EDU
Department of Electrical and Computer Engineering
University of Minnesota Twin Cities
Minneapolis, MN, 55455, USA
Tuo Zhao† TZHAO5@JHU.EDU
Department of Computer Science
Johns Hopkins University
Baltimore, MD, 21210, USA
Xiaoming Yuan XMYUAN@HKBU.EDU.HK
Department of Mathematics
Hong Kong Baptist University
Hong Kong, China
Han Liu HANLIU@PRINCETON.EDU
Department of Operations Research and Financial Engineering
Princeton University
Princeton, NJ 08544, USA

Editor: Mikio Braun

Abstract

This paper describes an `R` package named `flare`, which implements a family of new high dimensional regression methods (LAD Lasso, SQRT Lasso, ℓ_q Lasso, and Dantzig selector) and their extensions to sparse precision matrix estimation (TIGER and CLIME). These methods exploit different nonsmooth loss functions to gain modeling flexibility, estimation robustness, and tuning insensitiveness. The developed solver is based on the alternating direction method of multipliers (ADMM). The package `flare` is coded in double precision `C`, and called from `R` by a user-friendly interface. The memory usage is optimized by using the sparse matrix output. The experiments show that `flare` is efficient and can scale up to large problems.

Keywords: sparse linear regression, sparse precision matrix estimation, alternating direction method of multipliers, robustness, tuning insensitiveness

1. Introduction

As a popular sparse linear regression method for high dimensional data analysis, Lasso has been extensively studied by machine learning scientists (Tibshirani, 1996). It adopts the ℓ_1-regularized least square formulation to select and estimate nonzero parameters simultaneously. Software packages such as `glmnet` and `huge` have been developed to efficiently

∗. The package vignette is an extended version of this paper, which contains more technical details.
†. Xingguo Li and Tuo Zhao contributed equally to this work.

solve large problems (Friedman et al., 2010; Zhao et al., 2012, 2014). Lasso further yields a wide range of research interests, and motivates many variants by exploiting nonsmooth loss functions to gain modeling flexibility, estimation robustness, and tuning insensitiveness (See more details in the package vignette, Zhao and Liu (2014); Liu et al. (2014a)). These nonsmooth loss functions pose a great challenge to computation. To the best of our knowledge, no efficient solver has been developed so far for these Lasso variants.

In this report, we describe a newly developed `R` package named `flare` (Family of Lasso Regression). The `flare` package implements a family of linear regression methods including: (1) LAD Lasso, which is robust to heavy tail random noise and outliers (Wang, 2013); (2) SQRT Lasso, which is tuning insensitive (the optimal regularization parameter selection does not depend on any unknown parameter, Belloni et al. (2011)); (3) ℓ_q Lasso, which shares the advantage of LAD Lasso and SQRT Lasso; (4) Dantzig selector, which can tolerate missing values in the design matrix and response vector (Candes and Tao, 2007). By adopting the column by column regression scheme, we further extend these regression methods to sparse precision matrix estimation, including: (5) TIGER, which is tuning insensitive (Liu and Wang, 2012); (6) CLIME, which can tolerate missing values in the data matrix (Cai et al., 2011). The developed solver is based on the alternating direction method of multipliers (ADMM), which is further accelerated by a multistage screening approach (Boyd et al., 2011; Liu et al., 2014b). The global convergence result of ADMM has been established in He and Yuan (2015, 2012). The numerical simulations show that the `flare` package is efficient and can scale up to large problems.

2. Algorithm

We are interested in solving convex programs in the following generic form

$$\widehat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta},\ \boldsymbol{\alpha}}{\operatorname{argmin}} L_\lambda(\boldsymbol{\alpha}) + \|\boldsymbol{\beta}\|_1 \quad \text{subject to } \boldsymbol{r} - \mathbf{A}\boldsymbol{\beta} = \boldsymbol{\alpha}. \tag{1}$$

where $\lambda > 0$ is the regularization parameter. The possible choices of $L_\lambda(\boldsymbol{\alpha})$, $\mathbf{A}$, and $\boldsymbol{r}$ for different regression methods are listed in Table 1. Note that LAD Lasso and SQRT Lasso are special examples of ℓ_q Lasso for $q = 1$ and $q = 2$ respectively.
All methods in Table 1 can be efficiently solved by the iterative scheme as follows

$$\boldsymbol{\alpha}^{t+1} = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \frac{1}{2} \left\|\boldsymbol{u}^t + \boldsymbol{r} - \mathbf{A}\boldsymbol{\beta}^t - \boldsymbol{\alpha}\right\|_2^2 + \frac{1}{\rho} L_\lambda(\boldsymbol{\alpha}), \tag{2}$$

$$\boldsymbol{\beta}^{t+1} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \frac{1}{2} \left\|\boldsymbol{u}^t - \boldsymbol{\alpha}^{t+1} + \boldsymbol{r} - \mathbf{A}\boldsymbol{\beta}\right\|_2^2 + \frac{1}{\rho} \|\boldsymbol{\beta}\|_1, \tag{3}$$

$$\boldsymbol{u}^{t+1} = \boldsymbol{u}^t + (\boldsymbol{r} - \boldsymbol{\alpha}^{t+1} - \mathbf{A}\boldsymbol{\beta}^{t+1}), \tag{4}$$

where $\boldsymbol{u}$ is the rescaled Lagrange multiplier (Boyd et al., 2011), and $\rho > 0$ is the penalty parameter. For LAD Lasso, SQRT Lasso, or Dantzig selector, (2) has a closed form solution via the winsorization, soft thresholding, and group soft thresholding operators respectively. For L_q Lasso with $1 < q < 2$, (2) can be solved by the bisection-based root finding algorithm. (3) is a Lasso problem, which can be (approximately) solved by linearization or coordinate descent. Besides the pathwise optimization scheme and the active set trick, we also adopt the multistage screening approach to speedup the computation. In particular, we first

select k nested subsets of coordinates $\mathcal{A}_1 \subseteq \mathcal{A}_2 \subseteq ... \subseteq \mathcal{A}_k = \mathbb{R}^d$ by the marginal correlation between the covariates and responses. Then the algorithm iterates over these nested subsets of coordinates to obtain the solution. The multistage screening approach can greatly boost the empirical performance, especially for Dantzig selector.

Method	Loss function	$\mathbf{A}$	$\boldsymbol{r}$	Existing solver
L_q Lasso	$L_\lambda(\boldsymbol{\alpha}) = \frac{1}{\sqrt[q]{n}\lambda}\|\boldsymbol{\alpha}\|_q$	$\mathbf{X}$	$\boldsymbol{y}$	L.P. or S.O.C.P.
Dantzig selector	$L_\lambda(\boldsymbol{\alpha}) = \begin{cases} \infty & \text{if } \|\boldsymbol{\alpha}\|_\infty > \lambda \\ 0 & \text{otherwise} \end{cases}$	$\frac{1}{n}\mathbf{X}^T\mathbf{X}$	$\frac{1}{n}\mathbf{X}^T\boldsymbol{y}$	L.P.

Table 1: All regression methods provided in the `flare` package. $\mathbf{X} \in \mathbb{R}^{n\times d}$ denotes the design matrix, and $\boldsymbol{y} \in \mathbb{R}^n$ denotes the response vector. "L.P." denotes the general linear programming solver, and "S.O.C.P" denotes the second-order cone programming solver.

3. Examples

We illustrate the user interface by analyzing the eye disease data set in `flare`.

```
> # Load the data set
> library(flare); data(eyedata)
> # SQRT Lasso
> out1 = slim(x,y,method="lq",nlambda=40,lambda.min.value=sqrt(log(200)/120))
> # Dantzig Selector
> out2 = slim(x,y,method="dantzig",nlambda=40,lambda.min.ratio=0.35)
```

The program automatically generates a sequence of 40 regularization parameters and estimates the corresponding solution paths of SQRT Lasso and the Dantzig selector. For the Dantzig selector, the optimal regularization parameter is usually selected based on some model selection procedures, such as cross validation. Note that Belloni et al. (2011) has shown that the theoretically consistent regularization parameter of SQRT Lasso is $C\sqrt{\log d}/n$, where C is some constant. Thus we manually choose its minimum regularization parameter to be $\sqrt{\log(d)/n} = \sqrt{\log(200)/120}$. The minimum regularization parameter yields 19 nonzero coefficients out of 200.

4. Numerical Simulation

All experiments below are carried out on a PC with Intel Core i5 3.3GHz processor, and the convergence threshold of `flare` is chosen to be 10^{-5}. Timings (in seconds) are averaged over 100 replications using 20 regularization parameters, and the range of regularization parameters is chosen so that each method produces approximately the same number of nonzero estimates.

We first evaluate the timing performance of `flare` for sparse linear regression. We set $n = 100$ and vary d from 375 to 3000 as is shown in Table 2. We independently generate

each row of the design matrix from a d-dimensional normal distribution $N(0, \mathbf{\Sigma})$, where $\mathbf{\Sigma}_{jk} = 0.5^{|j-k|}$. Then we generate the response vector using $y_i = 3\mathbf{X}_{i1} + 2\mathbf{X}_{i2} + 1.5\mathbf{X}_{i4} + \epsilon_i$, where ϵ_i is independently generated from $N(0, 1)$. From Table 2, we see that all methods achieve good timing performance. Dantzig selector and ℓ_q Lasso are slower than the others due to more difficult computational formulations.

We then evaluate the timing performance of `flare` for sparse precision matrix estimation. We set $n = 100$ and vary d from 100 to 400 as is shown in Table 2. We independently generate the data from a d-dimensional normal distribution $N(0, \mathbf{\Sigma})$, where $\mathbf{\Sigma}_{jk} = 0.5^{|j-k|}$. The corresponding precision matrix $\mathbf{\Omega} = \mathbf{\Sigma}^{-1}$ has $\mathbf{\Omega}_{jj} = 1.3333$, $\mathbf{\Omega}_{jk} = -0.6667$ for all $j, k = 1, ..., d$ and $|j - k| = 1$, and all other entries are 0. From Table 2, we see that both TIGER and CLIME achieve good timing performance, and CLIME is slower than TIGER due to a more difficult computational formulation.

Sparse Linear Regression				
Method	$d = 375$	$d = 750$	$d = 1500$	$d = 3000$
LAD Lasso	1.1713(0.2915)	1.1046(0.3640)	1.8103(0.2919)	3.1378(0.7753)
SQRT Lasso	0.4888(0.0264)	0.7330(0.1234)	0.9485(0.2167)	1.2761(0.1510)
$\ell_{1.5}$ Lasso	12.995(0.5535)	14.071(0.5966)	14.382(0.7390)	16.936(0.5696)
Dantzig selector	0.3245(0.1871)	1.5360(1.8566)	4.4669(5.9929)	17.034(23.202)
Sparse Precision Matrix Estimation				
Method	$d = 100$	$d = 200$	$d = 300$	d=400
TIGER	1.0637(0.0361)	4.6251(0.0807)	7.1860(0.0795)	11.085(0.1715)
CLIME	2.5761(0.3807)	20.137(3.2258)	42.882(18.188)	112.50(11.561)

Table 2: Average timing performance (in seconds) with standard errors in the parentheses on sparse linear regression and sparse precision matrix estimation.

5. Discussion and Conclusions

Though the `glmnet` package cannot handle nonsmooth loss functions, it is much faster than `flare` for solving Lasso,[1] and the `glmnet` package can also be applied to solve ℓ_1 regularized generalized linear model estimation problems, which `flare` cannot. Overall speaking, the `flare` package serves as an efficient complement to the `glmnet` package for high dimensional data analysis. We will continue to maintain and support this package.

Acknowledgments

Tuo Zhao and Han Liu are supported by NSF Grants III-1116730 and NSF III-1332109, NIH R01MH102339, NIH R01GM083084, and NIH R01HG06841, and FDA HHSF2232 01000072C. Xiaoming Yuan is supported by the General Research Fund form Hong Kong Research Grants Council: 203311 and 203712.

1. See more detail in the package vignette.

References

A. Belloni, V. Chernozhukov, and L. Wang. Square-root lasso: pivotal recovery of sparse signals via conic programming. *Biometrika*, 98(4):791–806, 2011.

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

T. Cai, W. Liu, and X. Luo. A constrained ℓ_1 minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association*, 106:594–607, 2011.

E. Candes and T. Tao. The dantzig selector: Statistical estimation when p is much larger than n. *The Annals of Statistics*, 35(6):2313–2351, 2007.

J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1, 2010.

B. He and X. Yuan. On the $O(1/n)$ convergence rate of the Douglas-Rachford alternating direction method. *SIAM Journal on Numerical Analysis*, 50(2):700–709, 2012.

B. He and X. Yuan. On non-ergodic convergence rate of Douglas-Rachford alternating direction method of multipliers. *Numerische Mathematik*, 2015. (Accepted).

H. Liu and L. Wang. Tiger: A tuning-insensitive approach for optimally estimating Gaussian graphical models. Technical report, Princeton University, 2012.

H. Liu, L. Wang, and T. Zhao. Multivariate regression with calibration. In *Advances in Neural Information Processing Systems*, pages 127–135, 2014a.

H. Liu, L. Wang, and T. Zhao. Sparse covariance matrix estimation with eigenvalue constraints. *Journal of Computational and Graphical Statistics*, 23(2):439–459, 2014b.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1996.

L. Wang. The L_1 penalized LAD estimator for high dimensional linear regression. *Journal of Multivariate Analysis*, 120:135–151, 2013.

T. Zhao and H. Liu. Calibrated precision matrix estimation for high-dimensional elliptical distributions. *IEEE Transactions on Information Theory*, 60(12):7874–7887, 2014.

T. Zhao, H. Liu, K. Roeder, J. Lafferty, and L. Wasserman. The huge package for high-dimensional undirected graph estimation in R. *The Journal of Machine Learning Research*, 13(1):1059–1062, 2012.

T. Zhao, H. Liu, and T. Zhang. A general theory of pathwise coordinate optimization. *arXiv preprint arXiv:1412.7477*, 2014.

Journal of Machine Learning Research 16 (2015) 559-616 Submitted 4/14; Revised 10/14; Published 3/15

Regularized M-estimators with Nonconvexity: Statistical and Algorithmic Theory for Local Optima

Po-Ling Loh LOH@WHARTON.UPENN.EDU
Department of Statistics
The Wharton School
466 Jon M. Huntsman Hall
3730 Walnut Street
Philadelphia, PA 19104, USA

Martin J. Wainwright WAINWRIG@BERKELEY.EDU
Departments of EECS and Statistics
263 Cory Hall
University of California
Berkeley, CA 94720, USA

Editor: Nicolai Meinshausen

Abstract

We provide novel theoretical results regarding local optima of regularized M-estimators, allowing for nonconvexity in both loss and penalty functions. Under restricted strong convexity on the loss and suitable regularity conditions on the penalty, we prove that *any stationary point* of the composite objective function will lie within statistical precision of the underlying parameter vector. Our theory covers many nonconvex objective functions of interest, including the corrected Lasso for errors-in-variables linear models; regression for generalized linear models with nonconvex penalties such as SCAD, MCP, and capped-ℓ_1; and high-dimensional graphical model estimation. We quantify statistical accuracy by providing bounds on the ℓ_1-, ℓ_2-, and prediction error between stationary points and the population-level optimum. We also propose a simple modification of composite gradient descent that may be used to obtain a near-global optimum within statistical precision ϵ_{stat} in $\log(1/\epsilon_{\text{stat}})$ steps, which is the fastest possible rate of any first-order method. We provide simulation studies illustrating the sharpness of our theoretical results.

Keywords: high-dimensional statistics, M-estimation, model selection, nonconvex optimization, nonconvex regularization

1. Introduction

Although recent years have brought about a flurry of work on optimization of convex functions, optimizing nonconvex functions is in general computationally intractable (Nesterov and Nemirovskii, 1987; Vavasis, 1995). Nonconvex functions may possess local optima that are not global optima, and iterative methods such as gradient or coordinate descent may terminate undesirably in local optima. Unfortunately, standard statistical results for nonconvex M-estimators often only provide guarantees for *global* optima. This leads to a significant gap between theory and practice, since computing global optima—or even near-global optima—in an efficient manner may be extremely difficult in practice. Nonetheless,

empirical studies have shown that local optima of various nonconvex M-estimators arising in statistical problems appear to be well-behaved (e.g., Breheny and Huang, 2011). This type of observation is the starting point of our work.

A key insight is that nonconvex functions occurring in statistics are not constructed adversarially, so that "good behavior" might be expected in practice. Our recent work (Loh and Wainwright, 2012) confirmed this intuition for one specific case: a modified version of the Lasso applicable to errors-in-variables regression. Although the Hessian of the modified objective has many negative eigenvalues in the high-dimensional setting, the objective function resembles a strongly convex function when restricted to a cone set that includes the stationary points of the objective. This allows us to establish bounds on the statistical and optimization error.

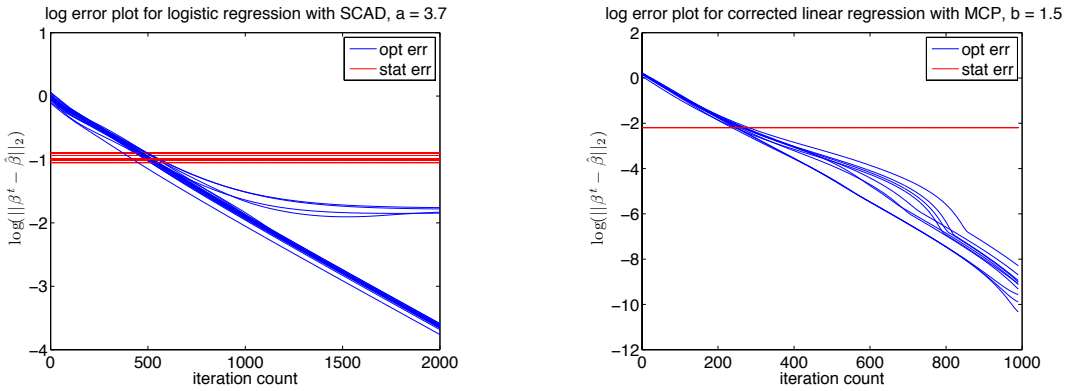
Our current paper is framed in a more general setting, and we focus on various M-estimators coupled with (nonconvex) regularizers of interest. On the statistical side, we establish bounds on the distance between *any local optimum* of the empirical objective and the unique minimizer of the population risk. Although the nonconvex functions may possess multiple local optima (as demonstrated in simulations), our theoretical results show that all local optima are essentially as good as a global optima from a statistical perspective. The results presented here subsume our previous work (Loh and Wainwright, 2012), and our present proof techniques are much more direct.

Our theory also sheds new light on a recent line of work involving the nonconvex SCAD and MCP regularizers (Fan and Li, 2001; Breheny and Huang, 2011; Zhang, 2010; Zhang and Zhang, 2012). Various methods previously proposed for nonconvex optimization include local quadratic approximation (LQA) (Fan and Li, 2001), minorization-maximization (MM) (Hunter and Li, 2005), local linear approximation (LLA) (Zou and Li, 2008), and coordinate descent (Breheny and Huang, 2011; Mazumder et al., 2011). However, these methods may terminate in local optima, which were not previously known to be well-behaved. In a recent paper, Zhang and Zhang (2012) provided statistical guarantees for global optima of least-squares linear regression with nonconvex penalties and showed that gradient descent starting from a Lasso solution would terminate in specific local minima. Fan et al. (2014) also showed that if the LLA algorithm is initialized at a Lasso optimum satisfying certain properties, the two-stage procedure produces an oracle solution for various nonconvex penalties. Finally, Chen and Gu (2014) showed that specific local optima of nonconvex regularized least-squares problems are stable, so optimization algorithms initialized sufficiently close by will converge to the same optima. See the survey paper (Zhang and Zhang, 2012) for a more complete overview of related work.

In contrast, our paper is the first to establish appropriate regularity conditions under which *all stationary points* (including both local and global optima) lie within a small ball of the population-level minimum. Thus, standard first-order methods such as projected and composite gradient descent (Nesterov, 2007) will converge to stationary points that lie within statistical error of the truth, eliminating the need for specially designed optimization algorithms that converge to specific local optima. Our work provides an important contribution to the growing literature on the tradeoff between statistical accuracy and optimization efficiency in high-dimensional problems, establishing that certain types of nonconvex M-estimators arising in statistical problems possess stationary points that both enjoy strong statistical guarantees and may be located efficiently. For a higher-level description of con-

temporary problems involving statistical and optimization tradeoffs, see Wainwright (2014) and the references cited therein.

Figure 1 provides an illustration of the type of behavior explained by the theory in this paper. Panel (a) shows the behavior of composite gradient descent for a form of logistic regression with the nonconvex SCAD (Fan and Li, 2001) as a regularizer: the red curve shows the *statistical error*, namely the ℓ_2-norm of the difference between a stationary point and the underlying true regression vector, and the blue curve shows the *optimization error*, meaning the difference between the iterates and a given global optimum. As shown by the blue curves, this problem possesses multiple local optima, since the algorithm converges to different final points depending on the initialization. However, as shown by the red curves, the statistical error of each local optimum is very low, so they are all essentially comparable from a statistical point of view. Panel (b) exhibits the same behavior for a problem in which



(a) (b)

Figure 1: Plots of the optimization error (blue curves) and statistical error (red curves) for a modified form of composite gradient descent, applicable to problems that may involve nonconvex cost functions and regularizers. (a) Plots for logistic regression with the nonconvex SCAD regularizer. (b) Plots for a corrected form of least squares (a nonconvex quadratic program) with the nonconvex MCP regularizer.

both the cost function (a corrected form of least-squares suitable for missing data, described in Loh and Wainwright, 2013a) and the regularizer (the MCP function, described in Zhang, 2010) are nonconvex. Nonetheless, as guaranteed by our theory, we still see the same qualitative behavior of the statistical and optimization error. Moreover, our theory also predicts the geometric convergence rates that are apparent in these plots. More precisely, under the same sufficient conditions for statistical consistency, we show that a modified form of composite gradient descent only requires $\log(1/\epsilon_{\text{stat}})$ steps to achieve a solution that is accurate up to the statistical precision ϵ_{stat}, which is the rate expected for *strongly convex* functions. Furthermore, our techniques are more generally applicable than the methods proposed by previous authors and are not restricted to least-squares or even convex loss functions.

While our paper was under review after its initial arXiv posting (Loh and Wainwright, 2013b), we became aware of an independent line of related work by Wang et al. (2014). Our contributions are substantially different, in that we provide sufficient conditions guaranteeing statistical consistency for *all* local optima, whereas their work is only concerned with establishing good behavior of successive iterates along a certain path-following algorithm. In addition, our techniques are applicable even to regularizers that do not satisfy smoothness constraints on the entire positive axis (such as capped-ℓ_1). Finally, we provide rigorous proofs showing the applicability of our sufficient condition on the loss function to a broad class of generalized linear models, whereas the applicability of their sparse eigenvalue condition to such objectives was not established.

The remainder of the paper is organized as follows. In Section 2, we set up basic notation and provide background on nonconvex regularizers and loss functions of interest. In Section 3, we provide our main theoretical results, including bounds on ℓ_1-, ℓ_2-, and prediction error, and also state corollaries for special cases. Section 4 contains a modification of composite gradient descent that may be used to obtain near-global optima and includes theoretical results establishing the linear convergence of our optimization algorithm. Section 5 supplies the results of various simulations. Proofs are contained in the Appendix. We note that a preliminary form of the results given here, without any proofs or algorithmic details, was presented at the NIPS conference (Loh and Wainwright, 2013c).

Notation: For functions $f(n)$ and $g(n)$, we write $f(n) \precsim g(n)$ to mean that $f(n) \leq cg(n)$ for some universal constant $c \in (0, \infty)$, and similarly, $f(n) \succsim g(n)$ when $f(n) \geq c'g(n)$ for some universal constant $c' \in (0, \infty)$. We write $f(n) \asymp g(n)$ when $f(n) \precsim g(n)$ and $f(n) \succsim g(n)$ hold simultaneously. For a vector $v \in \mathbb{R}^p$ and a subset $S \subseteq \{1, \ldots, p\}$, we write $v_S \in \mathbb{R}^S$ to denote the vector v restricted to S. For a matrix M, we write $|\!|\!| M |\!|\!|_2$ and $|\!|\!| M |\!|\!|_F$ to denote the spectral and Frobenius norms, respectively, and write $|\!|\!| M |\!|\!|_{\max} := \max_{i,j} |m_{ij}|$ to denote the elementwise ℓ_∞-norm of M. For a function $h : \mathbb{R}^p \to \mathbb{R}$, we write ∇h to denote a gradient or subgradient, if it exists. Finally, for $q, r > 0$, let $\mathbb{B}_q(r)$ denote the ℓ_q-ball of radius r centered around 0. We use the term "with high probability" (w.h.p.) to refer to events that occur with probability tending to 1 as $n, p, k \to \infty$. This is a loose requirement, but we will always take care to write out the expression for the probability explicitly (up to constant factors) in the formal statements of our theorems and corollaries below.

2. Problem Formulation

In this section, we develop some general theory for regularized M-estimators. We begin by establishing our notation and basic assumptions, before turning to the class of nonconvex regularizers and nonconvex loss functions to be covered in this paper.

2.1 Background

Given a collection of n samples $Z_1^n = \{Z_1, \ldots, Z_n\}$, drawn from a marginal distribution $\mathbb{P}$ over a space $\mathcal{Z}$, consider a loss function $\mathcal{L}_n : \mathbb{R}^p \times (\mathcal{Z})^n \to \mathbb{R}$. The value $\mathcal{L}_n(\beta; Z_1^n)$ serves as a measure of the "fit" between a parameter vector $\beta \in \mathbb{R}^p$ and the observed data. This empirical loss function should be viewed as a surrogate to the *population risk function*

$\mathcal{L} : \mathbb{R}^p \to \mathbb{R}$, given by

$$\mathcal{L}(\beta) := \mathbb{E}_Z\big[\mathcal{L}_n(\beta; Z_1^n)\big].$$

Our goal is to estimate the parameter vector $\beta^* := \arg\min_{\beta \in \mathbb{R}^p} \mathcal{L}(\beta)$ that minimizes the population risk, assumed to be unique.

To this end, we consider a regularized M-estimator of the form

$$\widehat{\beta} \in \arg\min_{g(\beta) \leq R,\, \beta \in \Omega} \left\{\mathcal{L}_n(\beta; Z_1^n) + \rho_\lambda(\beta)\right\}, \tag{1}$$

where $\rho_\lambda : \mathbb{R}^p \to \mathbb{R}$ is a *regularizer*, depending on a tuning parameter $\lambda > 0$, which serves to enforce a certain type of structure on the solution. Here, $R > 0$ is another tuning parameter that much be chosen carefully to make β^* a feasible point. In all cases, we consider regularizers that are separable across coordinates, and with a slight abuse of notation, we write

$$\rho_\lambda(\beta) = \sum_{j=1}^{p} \rho_\lambda(\beta_j).$$

Our theory allows for possible nonconvexity in *both* the loss function $\mathcal{L}_n$ and the regularizer ρ_λ. Due to this potential nonconvexity, our M-estimator also includes a side constraint $g : \mathbb{R}^p \to \mathbb{R}_+$, which we require to be a convex function satisfying the lower bound $g(\beta) \geq \|\beta\|_1$ for all $\beta \in \mathbb{R}^p$. Consequently, any feasible point for the optimization problem (1) satisfies the constraint $\|\beta\|_1 \leq R$, and as long as the empirical loss and regularizer are continuous, the Weierstrass extreme value theorem guarantees that a global minimum $\widehat{\beta}$ exists. Finally, our theory also allows for an additional side constraint of the form $\beta \in \Omega$, where Ω is some convex set containing β^*. For the graphical Lasso considered in Section 3.4, we take $\Omega = \mathcal{S}_+$ to be the set of positive semidefinite matrices; in settings where such an additional condition is extraneous, we simply set $\Omega = \mathbb{R}^p$.

2.2 Nonconvex Regularizers

We now state and discuss conditions on the regularizer, defined in terms of a univariate function $\rho_\lambda : \mathbb{R} \to \mathbb{R}$.

Assumption 1

(i) The function ρ_λ satisfies $\rho_\lambda(0) = 0$ and is symmetric around zero (i.e., $\rho_\lambda(t) = \rho_\lambda(-t)$ for all $t \in \mathbb{R}$).

(ii) On the nonnegative real line, the function ρ_λ is nondecreasing.

(iii) For $t > 0$, the function $t \mapsto \frac{\rho_\lambda(t)}{t}$ is nonincreasing in t.

(iv) The function ρ_λ is differentiable for all $t \neq 0$ and subdifferentiable at $t = 0$, with $\lim_{t \to 0^+} \rho'_\lambda(t) = \lambda L$.

(v) There exists $\mu > 0$ such that $\rho_{\lambda,\mu}(t) := \rho_\lambda(t) + \frac{\mu}{2} t^2$ is convex.

It is instructive to compare the conditions of Assumption 1 to similar conditions previously proposed in literature. Conditions (i)–(iii) are the same as those proposed in Zhang and Zhang (2012), except we omit the extraneous condition of subadditivity (cf. Lemma 1 of Chen and Gu, 2014). Such conditions are relatively mild and are satisfied for a wide variety of regularizers. Condition (iv) restricts the class of penalties by excluding regularizers such as the bridge (ℓ_q-) penalty, which has infinite derivative at 0; and the capped-ℓ_1 penalty, which has points of non-differentiability on the positive real line. However, one may check that if ρ_λ has an unbounded derivative at zero, then $\widetilde{\beta} = 0$ is *always* a local optimum of the composite objective (1), so there is no hope for $\|\widetilde{\beta} - \beta^*\|_2$ to be vanishingly small. Condition (v), known as *weak convexity* (Vial, 1982), also appears in Chen and Gu (2014) and is a type of curvature constraint that controls the level of nonconvexity of ρ_λ. Although this condition is satisfied by many regularizers of interest, it is again not satisfied by capped-ℓ_1 for any $\mu > 0$. For details on how our arguments may be modified to handle the more tricky capped-ℓ_1 penalty, see Appendix F.

Nonetheless, many regularizers that are commonly used in practice satisfy all the conditions in Assumption 1. It is easy to see that the standard ℓ_1-norm $\rho_\lambda(\beta) = \lambda\|\beta\|_1$ satisfies these conditions. More exotic functions have been studied in a line of past work on nonconvex regularization, and we provide a few examples here:

SCAD penalty: This penalty, due to Fan and Li (2001), takes the form

$$\rho_\lambda(t) := \begin{cases} \lambda|t|, & \text{for } |t| \leq \lambda, \\ -(t^2 - 2a\lambda|t| + \lambda^2)/(2(a-1)), & \text{for } \lambda < |t| \leq a\lambda, \\ (a+1)\lambda^2/2, & \text{for } |t| > a\lambda, \end{cases} \tag{2}$$

where $a > 2$ is a fixed parameter. As verified in Lemma 6 of Appendix A.2, the SCAD penalty satisfies the conditions of Assumption 1 with $L = 1$ and $\mu = \frac{1}{a-1}$.

MCP regularizer: This penalty, due to Zhang (2010), takes the form

$$\rho_\lambda(t) := \operatorname{sign}(t)\, \lambda \cdot \int_0^{|t|} \Big(1 - \frac{z}{\lambda b}\Big)_+ dz, \tag{3}$$

where $b > 0$ is a fixed parameter. As verified in Lemma 7 in Appendix A.2, the MCP regularizer satisfies the conditions of Assumption 1 with $L = 1$ and $\mu = \frac{1}{b}$.

2.3 Nonconvex Loss Functions and Restricted Strong Convexity

Throughout this paper, we require the loss function $\mathcal{L}_n$ to be differentiable, but we do not require it to be convex. Instead, we impose a weaker condition known as restricted strong convexity (RSC). Such conditions have been discussed in previous literature (Negahban et al., 2012; Agarwal et al., 2012), and involve a lower bound on the remainder in the first-order Taylor expansion of $\mathcal{L}_n$. In particular, our main statistical result is based on the

following RSC condition:

$$\langle \nabla \mathcal{L}_n(\beta^* + \Delta) - \nabla \mathcal{L}_n(\beta^*), \Delta \rangle \geq \begin{cases} \alpha_1 \|\Delta\|_2^2 - \tau_1 \dfrac{\log p}{n} \|\Delta\|_1^2, & \forall \|\Delta\|_2 \leq 1, \quad (4a) \\ \alpha_2 \|\Delta\|_2 - \tau_2 \sqrt{\dfrac{\log p}{n}} \|\Delta\|_1, & \forall \|\Delta\|_2 \geq 1, \quad (4b) \end{cases}$$

where the α_j's are strictly positive constants and the τ_j's are nonnegative constants.

To understand this condition, note that if $\mathcal{L}_n$ were actually strongly convex, then both these RSC inequalities would hold with $\alpha_1 = \alpha_2 > 0$ and $\tau_1 = \tau_2 = 0$. However, in the high-dimensional setting ($p \gg n$), the empirical loss $\mathcal{L}_n$ will not in general be strongly convex or even convex, but the RSC condition may still hold with strictly positive (α_j, τ_j). In fact, if $\mathcal{L}_n$ is convex (but not strongly convex), the left-hand expression in (4) is always nonnegative, so (4a) and (4b) hold trivially for $\frac{\|\Delta\|_1}{\|\Delta\|_2} \geq \sqrt{\frac{\alpha_1 n}{\tau_1 \log p}}$ and $\frac{\|\Delta\|_1}{\|\Delta\|_2} \geq \frac{\alpha_2}{\tau_2}\sqrt{\frac{n}{\log p}}$, respectively. Hence, the RSC inequalities only enforce a type of strong convexity condition over a cone of the form $\left\{ \frac{\|\Delta\|_1}{\|\Delta\|_2} \leq c\sqrt{\frac{n}{\log p}} \right\}$.

It is important to note that the class of functions satisfying RSC conditions of this type is much larger than the class of convex functions; for instance, our own past work (Loh and Wainwright, 2012) exhibits a large family of nonconvex quadratic functions that satisfy the condition (see Section 3.2 below for further discussion). Furthermore, note that we have stated two separate RSC inequalities (4) for different ranges of $\|\Delta\|_2$, unlike in past work (Negahban et al., 2012; Agarwal et al., 2012; Loh and Wainwright, 2012). As illustrated in the corollaries of Sections 3.3 and 3.4 below, an equality of the first type (4a) will only hold locally over Δ when we have more complicated types of loss functions that are only quadratic around a neighborhood of the origin. As proved in Appendix B.1, however, (4b) is implied by (4a) in cases when $\mathcal{L}_n$ is convex, which sustains our theoretical conclusions even under the weaker RSC conditions (4). Further note that by the inequality

$$\mathcal{L}_n(\beta^* + \Delta) - \mathcal{L}_n(\beta^*) \leq \langle \nabla \mathcal{L}_n(\beta^* + \Delta), \Delta \rangle,$$

which holds whenever $\mathcal{L}_n$ is convex, the RSC condition appearing in past work (e.g., Agarwal et al., 2012) implies that (4a) holds, so (4b) also holds by Lemma 8 in Appendix B.1. In cases where $\mathcal{L}_n$ is quadratic but not necessarily convex (cf. Section 3.2), our RSC condition (4) is again no stronger than the conditions appearing in past work, since those RSC conditions enforce (4a) globally over $\Delta \in \mathbb{R}^p$, which by Lemma 9 in Appendix B.1 implies that (4b) holds, as well. To allow for more general situations where $\mathcal{L}_n$ may be non-quadratic and/or nonconvex, we prefer to use the RSC formulation (4) in this paper.

Finally, we clarify that whereas Negahban et al. (2012) define an RSC condition with respect to a fixed subset $S \subseteq \{1, \ldots, p\}$, we follow the setup of Agarwal et al. (2012) and Loh and Wainwright (2012) and essentially require an RSC condition of the type defined in Negahban et al. (2012) to hold uniformly over *all* subsets S of size k. Although the results on statistical consistency may be established under the weaker RSC assumption with $S := \text{supp}(\beta^*)$, a uniform RSC condition is preferred because the true support set is not known a priori. The uniform RSC condition may be shown to hold w.h.p. in the sub-Gaussian settings we consider here (cf. Sections 3.2—3.4 below); in fact, the proofs contained in Negahban et al. (2012) establish a uniform RSC condition, as well.

3. Statistical Guarantees and Consequences

With this setup, we now turn to the statements and proofs of our main statistical guarantees, as well as some consequences for various statistical models. Our theory applies to any vector $\widetilde{\beta} \in \mathbb{R}^p$ that satisfies the *first-order necessary conditions* to be a local minimum of the program (1):

$$\langle \nabla \mathcal{L}_n(\widetilde{\beta}) + \nabla \rho_\lambda(\widetilde{\beta}),\, \beta - \widetilde{\beta} \rangle \geq 0, \qquad \text{for all feasible } \beta \in \mathbb{R}^p. \tag{5}$$

When $\widetilde{\beta}$ lies in the interior of the constraint set, this condition reduces to the usual zero-subgradient condition:

$$\nabla \mathcal{L}_n(\widetilde{\beta}) + \nabla \rho_\lambda(\widetilde{\beta}) = 0.$$

Such vectors $\widetilde{\beta}$ satisfying the condition (5) are also known as *stationary points* (Bertsekas, 1999); note that the set of stationary points also includes interior local maxima. Hence, although some of the discussion below is stated in terms of "local minima," the results hold for interior local maxima, as well.

3.1 Main Statistical Results

Our main theorems are deterministic in nature and specify conditions on the regularizer, loss function, and parameters that guarantee that any local optimum $\widetilde{\beta}$ lies close to the target vector $\beta^* = \arg\min_{\beta \in \mathbb{R}^p} \mathcal{L}(\beta)$. Corresponding probabilistic results will be derived in subsequent sections, where we establish that for appropriate choices of parameters (λ, R), the required conditions hold with high probability. Applying the theorems to particular models requires bounding the random quantity $\|\nabla \mathcal{L}_n(\beta^*)\|_\infty$ and verifying the RSC conditions (4). We begin with a theorem that provides guarantees on the error $\widetilde{\beta} - \beta^*$ as measured in the ℓ_1- and ℓ_2-norms:

Theorem 1 *Suppose the regularizer ρ_λ satisfies Assumption 1, the empirical loss $\mathcal{L}_n$ satisfies the RSC conditions* (4) *with $\frac{3}{4}\mu < \alpha_1$, and β^* is feasible for the objective. Consider any choice of λ such that*

$$\frac{4}{L} \cdot \max\left\{ \|\nabla \mathcal{L}_n(\beta^*)\|_\infty,\, \alpha_2 \sqrt{\frac{\log p}{n}} \right\} \;\leq\; \lambda \;\leq\; \frac{\alpha_2}{6RL}, \tag{6}$$

and suppose $n \geq \frac{16R^2 \max(\tau_1^2, \tau_2^2)}{\alpha_2^2} \log p$. Then any vector $\widetilde{\beta}$ satisfying the first-order necessary conditions (5) *satisfies the error bounds*

$$\|\widetilde{\beta} - \beta^*\|_2 \leq \frac{6\lambda L \sqrt{k}}{4\alpha_1 - 3\mu}, \qquad \text{and} \qquad \|\widetilde{\beta} - \beta^*\|_1 \leq \frac{24\lambda L k}{4\alpha_1 - 3\mu}, \tag{7}$$

where $k = \|\beta^*\|_0$.

From the bound (7), note that the squared ℓ_2-error grows proportionally with k, the number of nonzeros in the target parameter, and with λ^2. As will be clarified in the following sections, choosing λ proportional to $\sqrt{\frac{\log p}{n}}$ and R proportional to $\frac{1}{\lambda}$ will satisfy

the requirements of Theorem 1 w.h.p. for many statistical models, in which case we have a squared-ℓ_2 error that scales as $\frac{k \log p}{n}$, as expected.

Our next theorem provides a bound on a measure of the prediction error, as defined by the quantity

$$D(\widetilde{\beta}; \beta^*) := \langle \nabla \mathcal{L}_n(\widetilde{\beta}) - \nabla \mathcal{L}_n(\beta^*), \, \widetilde{\beta} - \beta^* \rangle. \tag{8}$$

When the empirical loss $\mathcal{L}_n$ is a convex function, this measure is always nonnegative, and in various special cases, it has a form that is readily interpretable. For instance, in the case of the least-squares objective function $\mathcal{L}_n(\beta) = \frac{1}{2n}\|y - X\beta\|_2^2$, we have

$$D(\widetilde{\beta}; \beta^*) = \frac{1}{n}\|X(\widetilde{\beta} - \beta^*)\|_2^2 = \frac{1}{n}\sum_{i=1}^{n} \big(\langle x_i, \, \widetilde{\beta} - \beta^* \rangle\big)^2,$$

corresponding to the usual measure of (fixed design) prediction error for a linear regression problem (cf. Corollary 1 below). More generally, when the loss function is the negative log likelihood for a generalized linear model with cumulant function ψ, the error measure (8) is equivalent to the symmetrized Bregman divergence defined by ψ. (See Section 3.3 for further details.)

Theorem 2 *Under the same conditions as Theorem 1, the error measure* (8) *is bounded as*

$$\langle \nabla \mathcal{L}_n(\widetilde{\beta}) - \nabla \mathcal{L}_n(\beta^*), \, \widetilde{\beta} - \beta^* \rangle \leq \lambda^2 L^2 k \left(\frac{9}{4\alpha_1 - 3\mu} + \frac{27\mu}{(4\alpha_1 - 3\mu)^2} \right). \tag{9}$$

This result shows that the prediction error (8) behaves similarly to the squared Euclidean norm between $\widetilde{\beta}$ and β^*.

Remark on (α_1, μ)**:** It is worthwhile to discuss the quantity $4\alpha_1 - 3\mu$ appearing in the denominator of the bounds in Theorems 1 and 2. Recall that α_1 measures the level of curvature of the loss function $\mathcal{L}_n$, while μ measures the level of nonconvexity of the penalty ρ_λ. Intuitively, the two quantities should play opposing roles in our result: larger values of μ correspond to more severe nonconvexity of the penalty, resulting in worse behavior of the overall objective (1), whereas larger values of α_1 correspond to more (restricted) curvature of the loss, leading to better behavior. However, while the condition $\frac{3}{4}\mu < \alpha_1$ is needed for the proof technique employed in Theorem 1, it does not seem to be strictly necessary in order to guarantee good behavior of local optima. As a careful examination of the proof reveals, the condition may be replaced by the alternate condition $c\mu < \alpha_1$, for any constant $c > \frac{1}{2}$. However, note that the capped-ℓ_1 penalty may be viewed as a limiting version of SCAD when $a \to 1$, or equivalently, $\mu \to \infty$. Viewed in this light, Theorem 4, to be stated and proved in Appendix F, reveals that a condition of the form $c\mu < \alpha_1$ is not necessary, at least in general, for good behavior of local optima. Moreover, Section 5 contains empirical studies using linear regression and the SCAD penalty showing that local optima may be well-behaved when $\alpha_1 < \frac{3}{4}\mu$. Nonetheless, our simulations (see Figure 5) also convey a cautionary message: In extreme cases, where α_1 is significantly smaller than μ, the good behavior of local optima (and the optimization algorithms used to find them) appear to degenerate.

Finally, we note that Negahban et al. (2012) have shown that for convex M-estimators, the arguments used to analyze ℓ_1-regularizers may be generalized to other types of "decomposable" regularizers, such as norms for group sparsity or the nuclear norm for low-rank matrices. In our present setting, where we allow for nonconvexity in the loss and regularizer, our theorems have straightforward and analogous generalizations.

We return to the proofs of Theorems 1 and 2 in Section 3.5. First, we develop various consequences of these theorems for various nonconvex loss functions and regularizers of interest. The main technical challenge is to establish that the RSC conditions (4) hold with high probability for appropriate choices of positive constants $\{(\alpha_j, \tau_j)\}_{j=1}^2$.

3.2 Corrected Linear Regression

We begin by considering the case of high-dimensional linear regression with systematically corrupted observations. Recall that in the framework of ordinary linear regression, we have the linear model

$$y_i = \underbrace{\langle \beta^*, x_i \rangle}_{\sum_{j=1}^p \beta_j^* x_{ij}} + \epsilon_i, \qquad \text{for } i = 1, \ldots, n, \tag{10}$$

where $\beta^* \in \mathbb{R}^p$ is the unknown parameter vector and $\{(x_i, y_i)\}_{i=1}^n$ are observations. Following a line of past work (e.g., Rosenbaum and Tsybakov, 2010; Loh and Wainwright, 2012), assume we instead observe pairs $\{(z_i, y_i)\}_{i=1}^n$, where the z_i's are systematically corrupted versions of the corresponding x_i's. Some examples of corruption mechanisms include the following:

(a) *Additive noise:* We observe $z_i = x_i + w_i$, where $w_i \in \mathbb{R}^p$ is a random vector independent of x_i, say zero-mean with known covariance matrix Σ_w.

(b) *Missing data:* For some fraction $\vartheta \in [0, 1)$, we observe a random vector $z_i \in \mathbb{R}^p$ such that for each component j, we independently observe $z_{ij} = x_{ij}$ with probability $1 - \vartheta$, and $z_{ij} = *$ with probability ϑ.

We use the population and empirical loss functions

$$\mathcal{L}(\beta) = \frac{1}{2}\beta^T \Sigma_x \beta - \beta^{*T} \Sigma_x \beta, \qquad \text{and} \qquad \mathcal{L}_n(\beta) = \frac{1}{2}\beta^T \widehat{\Gamma} \beta - \widehat{\gamma}^T \beta, \tag{11}$$

where $(\widehat{\Gamma}, \widehat{\gamma})$ are estimators for $(\Sigma_x, \Sigma_x \beta^*)$ that depend only on $\{(z_i, y_i)\}_{i=1}^n$. It is easy to see that $\beta^* = \arg\min_\beta \mathcal{L}(\beta)$. From the formulation (1), the corrected linear regression estimator is given by

$$\widehat{\beta} \in \arg \min_{g(\beta) \leq R} \left\{ \frac{1}{2}\beta^T \widehat{\Gamma} \beta - \widehat{\gamma}^T \beta + \rho_\lambda(\beta) \right\}. \tag{12}$$

We now state a concrete corollary in the case of additive noise (model (a) above). In this case, as discussed in Loh and Wainwright (2012), an appropriate choice of the pair $(\widehat{\Gamma}, \widehat{\gamma})$ is given by

$$\widehat{\Gamma} = \frac{Z^T Z}{n} - \Sigma_w, \qquad \text{and} \qquad \widehat{\gamma} = \frac{Z^T y}{n}. \tag{13}$$

Here, we assume the noise covariance Σ_w is known or may be estimated from replicates of the data. Such an assumption also appears in canonical errors-in-variables literature (Carroll et al., 1995), but it is an open question how to devise a corrected estimator when an estimate of Σ_w is not readily available. If we assume a sub-Gaussian model on the covariates and errors (i.e., x_i, w_i, and ϵ_i are sub-Gaussian with parameters σ_x^2, σ_w^2, and σ_ϵ^2, respectively), the contribution of the error covariances may be summarized in the error term

$$\varphi = (\sigma_x + \sigma_w)\big(\sigma_\epsilon + \sigma_w \|\beta^*\|_2\big), \tag{14}$$

which appears as a prefactor in the deviation bounds and estimation/prediction error bounds for the subsequent estimators (cf. Lemma 2 in Loh and Wainwright, 2012). We make this dependence explicit in the statement of the corollary for high-dimensional errors-in-variables regression below. Note in particular that φ scales up with both σ_ϵ and σ_w. Hence, even when $\sigma_\epsilon = 0$, corresponding to no additive error, we will have $\varphi \neq 0$ due to errors in the covariates; whereas when $\sigma_w = 0$, corresponding to cleanly observed covariates, we will still have $\varphi \neq 0$ due to the additional additive error introduced by the ϵ_i's, agreeing with canonical results for the Lasso (Bickel et al., 2009).

In the high-dimensional setting ($p \gg n$), the matrix $\widehat{\Gamma}$ in (13) is always negative definite: the matrix $\frac{Z^T Z}{n}$ has rank at most n, and the positive definite matrix Σ_w is then subtracted to obtain $\widehat{\Gamma}$. Consequently, the empirical loss function $\mathcal{L}_n$ previously defined (11) is nonconvex. Other choices of $\widehat{\Gamma}$ are applicable to missing data (model (b)), and also lead to nonconvex programs (see Loh and Wainwright, 2012 for further details).

Corollary 1 *Suppose we have i.i.d. observations $\{(z_i, y_i)\}_{i=1}^n$ from a corrupted linear model with additive noise, where the covariates and error terms are sub-Gaussian. Let φ be defined as in* (14) *with respect to the sub-Gaussian parameters. Suppose (λ, R) are chosen such that β^* is feasible and*

$$c\varphi\sqrt{\frac{\log p}{n}} \leq \lambda \leq \frac{c'}{R}.$$

Also suppose $\frac{3}{4}\mu < \frac{1}{2}\lambda_{\min}(\Sigma_x)$. Then given a sample size $n \geq C \max\{R^2, k\} \log p$, any stationary point $\widetilde{\beta}$ of the nonconvex program (12) *satisfies the estimation error bounds*

$$\|\widetilde{\beta} - \beta^*\|_2 \leq \frac{c_0 \lambda \sqrt{k}}{2\lambda_{\min}(\Sigma_x) - 3\mu}, \qquad \textit{and} \qquad \|\widetilde{\beta} - \beta^*\|_1 \leq \frac{c_0' \lambda k}{2\lambda_{\min}(\Sigma_x) - 3\mu},$$

and the prediction error bound

$$\widetilde{\nu}^T \widehat{\Gamma} \widetilde{\nu} \leq \lambda^2 k \left(\frac{\widetilde{c_0}}{2\lambda_{\min}(\Sigma_x) - 3\mu} + \frac{\widetilde{c_0}' \mu}{(2\lambda_{\min}(\Sigma_x) - 3\mu)^2} \right),$$

with probability at least $1 - c_1 \exp(-c_2 \log p)$, where $\|\beta^*\|_0 = k$.

When $\rho_\lambda(\beta) = \lambda\|\beta\|_1$ and $g(\beta) = \|\beta\|_1$, taking $\lambda \asymp \varphi\sqrt{\frac{\log p}{n}}$ and $R = b_0\sqrt{k}$ for some constant $b_0 \geq \|\beta^*\|_2$ yields the required scaling $n \gtrsim k \log p$. Hence, the bounds of Corollary 1 agree with bounds previously established in Theorem 1 of Loh and Wainwright (2012). Note, however, that those results are stated only for a *global minimum* $\widehat{\beta}$ of the program (12),

whereas Corollary 1 is a much stronger result holding for *any stationary point* $\widetilde{\beta}$. Theorem 2 of our earlier paper (Loh and Wainwright, 2012) provides a rather indirect (algorithmic) route for establishing similar bounds on $\|\widetilde{\beta} - \beta^*\|_1$ and $\|\widetilde{\beta} - \beta^*\|_2$, since the proposed projected gradient descent algorithm may become stuck at a stationary point. In contrast, our argument here is much more direct and does not rely on an algorithmic proof. Furthermore, our result is applicable to a more general class of (possibly nonconvex) penalties beyond the usual ℓ_1-norm.

Corollary 1 also has important consequences in the case where pairs $\{(x_i, y_i)\}_{i=1}^n$ from the linear model (10) are observed cleanly without corruption and ρ_λ is a nonconvex penalty. In that case, the empirical loss $\mathcal{L}_n$ previously defined (11) is equivalent to the least-squares loss, modulo a constant factor. Much existing work, including that of Fan and Li (2001) and Zhang and Zhang (2012), first establishes statistical consistency results concerning *global* minima of the program (12), then provides specialized algorithms such as a local linear approximation (LLA) for obtaining specific local optima that are provably close to the global optima. However, our results show that *any* optimization algorithm guaranteed to converge to a stationary point of the program suffices. See Section 4 for a more detailed discussion of optimization procedures and fast convergence guarantees for obtaining stationary points. In the fully-observed case, we also have $\widehat{\Gamma} = \frac{X^T X}{n}$, so the prediction error bound in Corollary 1 agrees with the familiar scaling $\frac{1}{n}\|X(\widetilde{\beta} - \beta^*)\|_2^2 \precsim \frac{k \log p}{n}$ appearing in ℓ_1-theory.

Furthermore, our theory provides a theoretical motivation for why the usual choice of $a = 3.7$ for linear regression with the SCAD penalty (Fan and Li, 2001) is reasonable. Indeed, as discussed in Section 2.2, we have

$$\mu = \frac{1}{a-1} \approx 0.37$$

in that case. Since $x_i \sim N(0, I)$ in the SCAD simulations, we have $\frac{3}{4}\mu < \frac{1}{2}\lambda_{\min}(\Sigma_x)$ for the choice $a = 3.7$. For further comments regarding the parameter a in the SCAD penalty, see the discussion concerning Figure 3 in Section 5.

3.3 Generalized Linear Models

Moving beyond linear regression, we now consider the case where observations are drawn from a generalized linear model (GLM). Recall that a GLM is characterized by the conditional distribution

$$\mathbb{P}(y_i \mid x_i, \beta, \sigma) = \exp\left\{\frac{y_i \langle \beta, x_i \rangle - \psi(x_i^T \beta)}{c(\sigma)}\right\},$$

where $\sigma > 0$ is a scale parameter and ψ is the cumulant function, By standard properties of exponential families (McCullagh and Nelder, 1989; Lehmann and Casella, 1998), we have

$$\psi'(x_i^T \beta) = \mathbb{E}[y_i \mid x_i, \beta, \sigma].$$

In our analysis, we assume that there exists $\alpha_u > 0$ such that $\psi''(t) \leq \alpha_u$, for all $t \in \mathbb{R}$. Note that this boundedness assumption holds in various settings, including linear regression, logistic regression, and multinomial regression, but does not hold for Poisson regression.

The bound will be necessary to establish both statistical consistency results in the present section and fast global convergence guarantees for our optimization algorithms in Section 4.

The population loss corresponding to the negative log likelihood is then given by

$$\mathcal{L}(\beta) = -\mathbb{E}[\log \mathbb{P}(x_i, y_i)] = -\mathbb{E}[\log \mathbb{P}(x_i)] - \frac{1}{c(\sigma)} \cdot \mathbb{E}[y_i \langle \beta, x_i \rangle - \psi(x_i^T \beta)],$$

giving rise to the population-level and empirical gradients

$$\nabla \mathcal{L}(\beta) = \frac{1}{c(\sigma)} \cdot \mathbb{E}[(\psi'(x_i^T \beta) - y_i) x_i], \quad \text{and}$$
$$\nabla \mathcal{L}_n(\beta) = \frac{1}{c(\sigma)} \cdot \frac{1}{n} \sum_{i=1}^n \big(\psi'(x_i^T \beta) - y_i\big) x_i.$$

Since we are optimizing over β, we will rescale the loss functions and assume $c(\sigma) = 1$. We may check that if β^* is the true parameter of the GLM, then $\nabla \mathcal{L}(\beta^*) = 0$; furthermore,

$$\nabla^2 \mathcal{L}_n(\beta) = \frac{1}{n} \sum_{i=1}^n \psi''(x_i^T \beta) x_i x_i^T \succeq 0,$$

so $\mathcal{L}_n$ is convex.

We will assume that β^* is sparse and optimize the penalized maximum likelihood program

$$\widehat{\beta} \in \arg \min_{g(\beta) \leq R} \left\{ \frac{1}{n} \sum_{i=1}^n \big(\psi(x_i^T \beta) - y_i x_i^T \beta\big) + \rho_\lambda(\beta) \right\}. \tag{15}$$

We then have the following corollary, proved in Appendix B.3:

Corollary 2 *Suppose we have i.i.d. observations $\{(x_i, y_i)\}_{i=1}^n$ from a GLM, where the x_i's are sub-Gaussian. Suppose (λ, R) are chosen such that β^* is feasible and*

$$c\sqrt{\frac{\log p}{n}} \leq \lambda \leq \frac{c'}{R}.$$

Then given a sample size $n \geq CR^2 \log p$, any stationary point $\widetilde{\beta}$ of the nonconvex program (15) satisfies

$$\|\widetilde{\beta} - \beta^*\|_2 \leq \frac{c_0 \lambda \sqrt{k}}{4\alpha_1 - 3\mu}, \qquad \text{and} \qquad \|\widetilde{\beta} - \beta^*\|_1 \leq \frac{c_0' \lambda k}{4\alpha_1 - 3\mu},$$

with probability at least $1 - c_1 \exp(-c_2 \log p)$, where $\|\beta^*\|_0 = k$. Here, α_1 is a constant depending on $\|\beta^*\|_2$, ψ, $\lambda_{\min}(\Sigma_x)$, and the sub-Gaussian parameter of the x_i's, and we assume $\mu < 2\alpha_1$.

Although $\mathcal{L}_n$ is convex in this case, the overall program may *not* be convex if the regularizer ρ_λ is nonconvex, giving rise to multiple local optima. For instance, see the simulations of Figure 4 in Section 5 for a demonstration of such local optima. In past work,

Breheny and Huang (2011) studied logistic regression with SCAD and MCP regularizers, but did not provide any theoretical results on the quality of the local optima. In this context, Corollary 2 shows that their coordinate descent algorithms are guaranteed to converge to a stationary point $\widetilde{\beta}$ within close proximity of the true parameter β^*.

In the statement of Corollary 2, we choose not to write out the form of α_1 explicitly as in Corollary 1, since it is rather complicated. As explained in the proof of Corollary 2 in Appendix B.3, the precise form of α_1 may be traced back to Proposition 2 of Negahban et al. (2012).

3.4 Graphical Lasso

Finally, we specialize our results to the case of the graphical Lasso. Given p-dimensional observations $\{x_i\}_{i=1}^n$, the goal is to estimate the structure of the underlying (sparse) graphical model. Recall that the population and empirical losses for the graphical Lasso are given by

$$\mathcal{L}(\Theta) = \text{trace}(\Sigma\Theta) - \log\det(\Theta), \quad \text{and} \quad \mathcal{L}_n(\Theta) = \text{trace}(\widehat{\Sigma}\Theta) - \log\det(\Theta),$$

where $\widehat{\Sigma}$ is an empirical estimate for the covariance matrix $\Sigma = \text{Cov}(x_i)$. The objective function for the graphical Lasso is then given by

$$\widehat{\Theta} \in \arg\min_{g(\Theta)\leq R,\, \Theta\succeq 0} \left\{ \text{trace}(\widehat{\Sigma}\Theta) - \log\det(\Theta) + \sum_{j,k=1}^{p} \rho_\lambda(\Theta_{jk}) \right\}, \tag{16}$$

where we apply the (possibly nonconvex) penalty function ρ_λ to all entries of Θ, and define $\Omega := \big\{\Theta \in \mathbb{R}^{p\times p} \mid \Theta = \Theta^T,\ \Theta \succeq 0\big\}$.

A host of statistical and algorithmic results have been established for the graphical Lasso in the case of Gaussian observations with an ℓ_1-penalty (Banerjee et al., 2008; Friedman et al., 2008; Rothman et al., 2008; Yuan and Lin, 2007), and more recently, for discrete-valued observations, as well (Loh and Wainwright, 2013a). In addition, a version of the graphical Lasso incorporating a nonconvex SCAD penalty has been proposed (Fan et al., 2009). Our results subsume previous Frobenius error bounds for the graphical Lasso and again imply that even in the presence of a nonconvex regularizer, all stationary points of the nonconvex program (16) remain close to the true inverse covariance matrix Θ^*.

As suggested by Loh and Wainwright (2013a), the graphical Lasso easily accommodates systematically corrupted observations, with the only modification being the form of the sample covariance matrix $\widehat{\Sigma}$. Just as in Corollary 1, the magnitude and form of corruption would occur as a prefactor in the deviation condition captured in (17) below; for instance, in the case of $\widehat{\Sigma} = \frac{Z^TZ}{n} - \Sigma_w$, corresponding to additive noise in the x_i's, the bound (17) would involve a prefactor of σ_z^2 rather than σ_x^2, where σ_z^2 and σ_x^2 are the sub-Gaussian parameters of z_i and x_i, respectively.

Further note that the program (16) is always useful for obtaining a consistent estimate of a sparse inverse covariance matrix, regardless of whether the x_i's are drawn from a distribution for which Θ^* is relevant in estimating the edges of the underlying graph. Note that other variants of the graphical Lasso exist in which only off-diagonal entries of Θ are penalized, and similar results for statistical consistency hold in that case. Here, we

assume that all entries are penalized equally in order to simplify our arguments. The same framework is considered by Fan et al. (2009).

We have the following result, proved in Appendix B.4. The statement of the corollary is purely deterministic, but in cases of interest (say, sub-Gaussian observations), the deviation condition (17) holds with probability at least $1 - c_1 \exp(-c_2 \log p)$, translating into the Frobenius norm bound (18) holding with the same probability.

Corollary 3 *Suppose we have an estimate $\widehat{\Sigma}$ of the covariance matrix Σ based on (possibly corrupted) observations $\{x_i\}_{i=1}^n$, such that*

$$\left|\!\left|\!\left| \widehat{\Sigma} - \Sigma \right|\!\right|\!\right|_{\max} \leq c_0 \sqrt{\frac{\log p}{n}}. \tag{17}$$

Also suppose Θ^* has at most s nonzero entries. Suppose (λ, R) are chosen such that Θ^* is feasible and

$$c\sqrt{\frac{\log p}{n}} \leq \lambda \leq \frac{c'}{R}.$$

Suppose $\frac{3}{4}\mu < (|||\Theta^*|||_2 + 1)^{-2}$. Then with a sample size $n > Cs \log p$, for a sufficiently large constant $C > 0$, any stationary point $\widetilde{\Theta}$ of the nonconvex program (16) satisfies

$$\left|\!\left|\!\left| \widetilde{\Theta} - \Theta^* \right|\!\right|\!\right|_F \leq \frac{c_0' \lambda \sqrt{s}}{4\left(|||\Theta^*|||_2 + 1\right)^{-2} - 3\mu}. \tag{18}$$

When ρ is simply the ℓ_1-penalty, the bound (18) from Corollary 3 matches the minimax rates for Frobenius norm estimation of an s-sparse inverse covariance matrix (Rothman et al., 2008; Ravikumar et al., 2011).

3.5 Proof of Theorems 1 and 2

We now turn to the proofs of our two main theorems.

Proof of Theorem 1: Introducing the shorthand $\widetilde{\nu} := \widetilde{\beta} - \beta^*$, we begin by proving that $\|\widetilde{\nu}\|_2 \leq 1$. If not, then (4b) gives the lower bound

$$\langle \nabla \mathcal{L}_n(\widetilde{\beta}) - \nabla \mathcal{L}_n(\beta^*), \widetilde{\nu} \rangle \geq \alpha_2 \|\widetilde{\nu}\|_2 - \tau_2 \sqrt{\frac{\log p}{n}} \|\widetilde{\nu}\|_1. \tag{19}$$

Since β^* is feasible, we may take $\beta = \beta^*$ in (5), and combining with (19) yields

$$\langle -\nabla \rho_\lambda(\widetilde{\beta}) - \nabla \mathcal{L}_n(\beta^*), \widetilde{\nu} \rangle \geq \alpha_2 \|\widetilde{\nu}\|_2 - \tau_2 \sqrt{\frac{\log p}{n}} \|\widetilde{\nu}\|_1. \tag{20}$$

By Hölder's inequality, followed by the triangle inequality, we also have

$$\begin{aligned} \langle -\nabla \rho_\lambda(\widetilde{\beta}) - \nabla \mathcal{L}_n(\beta^*), \widetilde{\nu} \rangle &\leq \left\{ \|\nabla \rho_\lambda(\widetilde{\beta})\|_\infty + \|\nabla \mathcal{L}_n(\beta^*)\|_\infty \right\} \|\widetilde{\nu}\|_1 \\ &\overset{(i)}{\leq} \left\{ \lambda L + \frac{\lambda L}{2} \right\} \|\widetilde{\nu}\|_1, \end{aligned}$$

where inequality (i) follows since $\|\nabla \mathcal{L}_n(\beta^*)\|_\infty \leq \frac{\lambda L}{2}$ by the bound (6), and $\|\nabla \rho_\lambda(\widetilde{\beta})\|_\infty \leq \lambda L$ by Lemma 4 in Appendix A.1. Combining this upper bound with (20) and rearranging then yields

$$\|\widetilde{\nu}\|_2 \leq \frac{\|\widetilde{\nu}\|_1}{\alpha_2}\left(\frac{3\lambda L}{2} + \tau_2\sqrt{\frac{\log p}{n}}\right) \leq \frac{2R}{\alpha_2}\left(\frac{3\lambda L}{2} + \tau_2\sqrt{\frac{\log p}{n}}\right).$$

By our choice of λ from (6) and the assumed lower bound on the sample size n, the right hand side is at most 1, so $\|\widetilde{\nu}\|_2 \leq 1$, as claimed.

Consequently, we may apply (4a), yielding the lower bound

$$\langle \nabla \mathcal{L}_n(\widetilde{\beta}) - \nabla \mathcal{L}_n(\beta^*), \widetilde{\nu}\rangle \geq \alpha_1 \|\widetilde{\nu}\|_2^2 - \tau_1 \frac{\log p}{n}\|\widetilde{\nu}\|_1^2. \tag{21}$$

Since the function $\rho_{\lambda,\mu}(\beta) := \rho_\lambda(\beta) + \frac{\mu}{2}\|\beta\|_2^2$ is convex by assumption, we have

$$\rho_{\lambda,\mu}(\beta^*) - \rho_{\lambda,\mu}(\widetilde{\beta}) \geq \langle \nabla \rho_{\lambda,\mu}(\widetilde{\beta}), \beta^* - \widetilde{\beta}\rangle \;=\; \langle \nabla \rho_\lambda(\widetilde{\beta}) + \mu\widetilde{\beta}, \beta^* - \widetilde{\beta}\rangle,$$

implying that

$$\langle \nabla \rho_\lambda(\widetilde{\beta}), \beta^* - \widetilde{\beta}\rangle \leq \rho_\lambda(\beta^*) - \rho_\lambda(\widetilde{\beta}) + \frac{\mu}{2}\|\widetilde{\beta} - \beta^*\|_2^2. \tag{22}$$

Combining (21) with (5) and (22), we obtain

$$\alpha_1\|\widetilde{\nu}\|_2^2 - \tau_1\frac{\log p}{n}\|\widetilde{\nu}\|_1^2 \leq -\langle \nabla \mathcal{L}_n(\beta^*), \widetilde{\nu}\rangle + \rho_\lambda(\beta^*) - \rho_\lambda(\widetilde{\beta}) + \frac{\mu}{2}\|\widetilde{\beta} - \beta^*\|_2^2.$$

Rearranging and using Hölder's inequality, we then have

$$\begin{aligned}\left(\alpha_1 - \frac{\mu}{2}\right)\|\widetilde{\nu}\|_2^2 &\leq \rho_\lambda(\beta^*) - \rho_\lambda(\widetilde{\beta}) + \|\nabla \mathcal{L}_n(\beta^*)\|_\infty \cdot \|\widetilde{\nu}\|_1 + \tau_1\frac{\log p}{n}\|\widetilde{\nu}\|_1^2 \\ &\leq \rho_\lambda(\beta^*) - \rho_\lambda(\widetilde{\beta}) + \left(\|\nabla \mathcal{L}_n(\beta^*)\|_\infty + 4R\tau_1\frac{\log p}{n}\right)\|\widetilde{\nu}\|_1. \end{aligned} \tag{23}$$

Note that by our assumptions, we have

$$\|\nabla \mathcal{L}_n(\beta^*)\|_\infty + 4R\tau_1\frac{\log p}{n} \leq \frac{\lambda L}{4} + \alpha_2\sqrt{\frac{\log p}{n}} \leq \frac{\lambda L}{2}.$$

Combining this with (23) and (53) in Lemma 4 in Appendix A.1, as well as the subadditivity of ρ_λ, we then have

$$\begin{aligned}\left(\alpha_1 - \frac{\mu}{2}\right)\|\widetilde{\nu}\|_2^2 &\leq \rho_\lambda(\beta^*) - \rho_\lambda(\widetilde{\beta}) + \frac{\lambda L}{2}\cdot\left(\frac{\rho_\lambda(\widetilde{\nu})}{\lambda L} + \frac{\mu}{2\lambda L}\|\widetilde{\nu}\|_2^2\right) \\ &\leq \rho_\lambda(\beta^*) - \rho_\lambda(\widetilde{\beta}) + \frac{\rho_\lambda(\beta^*) + \rho_\lambda(\widetilde{\beta})}{2} + \frac{\mu}{4}\|\widetilde{\nu}\|_2^2,\end{aligned}$$

implying that

$$0 \leq \left(\alpha_1 - \frac{3\mu}{4}\right) \|\widetilde{\nu}\|_2^2 \leq \frac{3}{2}\rho_\lambda(\beta^*) - \frac{1}{2}\rho_\lambda(\widetilde{\beta}). \tag{24}$$

In particular, we have $3\rho_\lambda(\beta^*) - \rho_\lambda(\widetilde{\beta}) \geq 0$, so we may apply Lemma 5 in Appendix A.1 to conclude that

$$3\rho_\lambda(\beta^*) - \rho_\lambda(\widetilde{\beta}) \leq 3\lambda L\|\widetilde{\nu}_A\|_1 - \lambda L\|\widetilde{\nu}_{A^c}\|_1, \tag{25}$$

where A denotes the index set of the k largest elements of $\widetilde{\beta} - \beta^*$ in magnitude. In particular, we have the cone condition

$$\|\widetilde{\nu}_{A^c}\|_1 \leq 3\|\widetilde{\nu}_A\|_1. \tag{26}$$

Substituting (25) into (24), we then have

$$\left(2\alpha_1 - \frac{3\mu}{2}\right) \|\widetilde{\nu}\|_2^2 \;\leq\; 3\lambda L\|\widetilde{\nu}_A\|_1 - \lambda L\|\widetilde{\nu}_{A^c}\|_1 \;\leq\; 3\lambda L\|\widetilde{\nu}_A\|_1 \;\leq\; 3\lambda L\sqrt{k}\|\widetilde{\nu}\|_2, \tag{27}$$

from which we conclude that

$$\|\widetilde{\nu}\|_2 \leq \frac{6\lambda L\sqrt{k}}{4\alpha_1 - 3\mu},$$

as wanted. The ℓ_1-bound follows from the ℓ_2-bound and the observation that

$$\|\widetilde{\nu}\|_1 \leq \|\widetilde{\nu}_A\|_1 + \|\widetilde{\nu}_{A^c}\|_1 \leq 4\|\widetilde{\nu}_A\|_1 \leq 4\sqrt{k}\|\widetilde{\nu}\|_2,$$

using the cone inequality (26).

Proof of Theorem 2: In order to establish (9), note that combining the first-order condition (5) with the upper bound (22), we have

$$\begin{aligned}\langle \nabla\mathcal{L}_n(\widetilde{\beta}) - \nabla\mathcal{L}_n(\beta^*), \widetilde{\nu}\rangle &\leq \langle -\nabla\rho_\lambda(\widetilde{\beta}) - \nabla\mathcal{L}_n(\beta^*), \widetilde{\nu}\rangle \\ &\leq \rho_\lambda(\beta^*) - \rho_\lambda(\widetilde{\beta}) + \frac{\mu}{2}\|\widetilde{\nu}\|_2^2 + \|\nabla\mathcal{L}_n(\beta^*)\|_\infty \cdot \|\widetilde{\nu}\|_1. \end{aligned} \tag{28}$$

Furthermore, as noted earlier, Lemma 4 in Appendix A.1 implies that

$$\|\nabla\mathcal{L}_n(\beta^*)\|_\infty \cdot \|\widetilde{\nu}\|_1 \leq \frac{\lambda L}{2} \cdot \left(\frac{\rho_\lambda(\beta^*) + \rho_\lambda(\widetilde{\beta})}{\lambda L} + \frac{\mu}{2\lambda L}\|\widetilde{\nu}\|_2^2\right) \leq \frac{\rho_\lambda(\beta^*) + \rho_\lambda(\widetilde{\beta})}{2} + \frac{\mu}{4}\|\widetilde{\nu}\|_2^2.$$

Substituting this into (28) then gives

$$\begin{aligned}\langle \nabla\mathcal{L}_n(\widetilde{\beta}) - \nabla\mathcal{L}_n(\beta^*), \widetilde{\nu}\rangle &\leq \frac{3}{2}\rho_\lambda(\beta^*) - \frac{1}{2}\rho_\lambda(\widetilde{\beta}) + \frac{3\mu}{4}\|\widetilde{\nu}\|_2^2 \\ &\leq \frac{3\lambda L}{2}\|\widetilde{\nu}_A\|_1 - \frac{\lambda L}{2}\|\widetilde{\nu}_{A^c}\|_1 + \frac{3\mu}{4}\|\widetilde{\nu}\|_2^2 \\ &\leq \frac{3\lambda L\sqrt{k}}{2}\|\widetilde{\nu}\|_2 + \frac{3\mu}{4}\|\widetilde{\nu}\|_2^2,\end{aligned}$$

so substituting in the ℓ_2-bound (7) yields the desired result.

4. Optimization Algorithms

We now describe how a version of composite gradient descent (Nesterov, 2007) may be applied to efficiently optimize the nonconvex program (1), and show that it enjoys a linear rate of convergence under suitable conditions. In this section, we focus exclusively on a version of the optimization problem with the side function

$$g_{\lambda,\mu}(\beta) := \frac{1}{\lambda}\Big\{\rho_\lambda(\beta) + \frac{\mu}{2}\|\beta\|_2^2\Big\}. \tag{29}$$

Note that this choice of $g_{\lambda,\mu}$ is convex by Assumption 1. We may then write the program (1) as

$$\widehat{\beta} \in \arg\min_{g_{\lambda,\mu}(\beta)\leq R,\ \beta\in\Omega} \Big\{ \underbrace{\big(\mathcal{L}_n(\beta) - \frac{\mu}{2}\|\beta\|_2^2\big)}_{\bar{\mathcal{L}}_n} + \lambda g_{\lambda,\mu}(\beta)\Big\}. \tag{30}$$

In this way, the objective function decomposes nicely into a sum of a differentiable but nonconvex function and a possibly nonsmooth but convex penalty. Applied to the representation (30) of the objective function, the composite gradient descent procedure of Nesterov (2007) produces a sequence of iterates $\{\beta^t\}_{t=0}^{\infty}$ via the updates

$$\beta^{t+1} \in \arg\min_{g_{\lambda,\mu}(\beta)\leq R,\ \beta\in\Omega} \left\{ \frac{1}{2}\left\|\beta - \left(\beta^t - \frac{\nabla\bar{\mathcal{L}}_n(\beta^t)}{\eta}\right)\right\|_2^2 + \frac{\lambda}{\eta} g_{\lambda,\mu}(\beta)\right\}, \tag{31}$$

where $\frac{1}{\eta}$ is the step size. As discussed in Section 4.2, these updates may be computed in a relatively straightforward manner.

4.1 Fast Global Convergence

The main result of this section is to establish that the algorithm defined by the iterates (31) converges very quickly to a δ-neighborhood of any global optimum, for all tolerances δ that are of the same order (or larger) than the statistical error.

We begin by setting up the notation and assumptions underlying our result. The *Taylor error* around the vector β_2 in the direction $\beta_1 - \beta_2$ is given by

$$\mathcal{T}(\beta_1, \beta_2) := \mathcal{L}_n(\beta_1) - \mathcal{L}_n(\beta_2) - \langle \nabla\mathcal{L}_n(\beta_2),\, \beta_1 - \beta_2\rangle. \tag{32}$$

We analogously define the Taylor error $\bar{\mathcal{T}}$ for the modified loss function $\bar{\mathcal{L}}_n$, and note that

$$\bar{\mathcal{T}}(\beta_1, \beta_2) = \mathcal{T}(\beta_1, \beta_2) - \frac{\mu}{2}\|\beta_1 - \beta_2\|_2^2. \tag{33}$$

For all vectors $\beta_2 \in \mathbb{B}_2(3) \cap \mathbb{B}_1(R)$, we require the following form of restricted strong convexity:

$$\mathcal{T}(\beta_1, \beta_2) \geq \begin{cases} \alpha_1\|\beta_1 - \beta_2\|_2^2 - \tau_1\dfrac{\log p}{n}\|\beta_1 - \beta_2\|_1^2, & \forall \|\beta_1 - \beta_2\|_2 \leq 3, \quad (34a) \\ \alpha_2\|\beta_1 - \beta_2\|_2 - \tau_2\sqrt{\dfrac{\log p}{n}}\|\beta_1 - \beta_2\|_1, & \forall \|\beta_1 - \beta_2\|_2 \geq 3. \quad (34b) \end{cases}$$

The conditions (34) are similar but not identical to the earlier RSC conditions (4). The main difference is that we now require the Taylor difference to be bounded below uniformly over $\beta_2 \in \mathbb{B}_2(3) \cap \mathbb{B}_1(R)$, as opposed to for a fixed $\beta_2 = \beta^*$. In addition, we assume an analogous upper bound on the Taylor series error:

$$\mathcal{T}(\beta_1, \beta_2) \leq \alpha_3 \|\beta_1 - \beta_2\|_2^2 + \tau_3 \frac{\log p}{n} \|\beta_1 - \beta_2\|_1^2, \qquad \text{for all } \beta_1, \beta_2 \in \Omega, \tag{35}$$

a condition referred to as *restricted smoothness* in past work (Agarwal et al., 2012). Throughout this section, we assume $2\alpha_i > \mu$ for all i, where μ is the coefficient ensuring the convexity of the function $g_{\lambda,\mu}$ from (29). Furthermore, we define $\alpha = \min\{\alpha_1, \alpha_2\}$ and $\tau = \max\{\tau_1, \tau_2, \tau_3\}$.

The following theorem applies to any population loss function $\mathcal{L}$ for which the population minimizer β^* is k-sparse and $\|\beta^*\|_2 \leq 1$. Similar results could be derived for general $\|\beta^*\|_2$, with the radius of the RSC condition (34a) replaced by $3\|\beta^*\|_2$ and Lemma 2 in Section 4.3 adjusted appropriately, but we only include the analysis for $\|\beta^*\|_2 \leq 1$ in order to simplify our exposition. We also assume the scaling $n > Ck \log p$, for a constant C depending on the α_i's and τ_i's. Note that this scaling is reasonable, since no estimator of a k-sparse vector in p dimensions can have low ℓ_2-error unless the condition holds (see Raskutti et al., 2011 for minimax rates). We show that the composite gradient updates (31) exhibit a type of *globally geometric convergence* in terms of the quantity

$$\kappa := \frac{1 - \frac{2\alpha - \mu}{8\eta} + \varphi(n, p, k)}{1 - \varphi(n, p, k)}, \qquad \text{where} \quad \varphi(n, p, k) := \frac{c\tau k \frac{\log p}{n}}{2\alpha - \mu}. \tag{36}$$

Under the stated scaling on the sample size, we are guaranteed that $\kappa \in (0, 1)$, so it is a *contraction factor*. Roughly speaking, we show that the squared optimization error will fall below δ^2 within $T \asymp \frac{\log(1/\delta^2)}{\log(1/\kappa)}$ iterations. More precisely, our theorem guarantees δ-accuracy for all iterations larger than

$$T^*(\delta) := \frac{2 \log\left(\frac{\phi(\beta^0) - \phi(\widehat{\beta})}{\delta^2}\right)}{\log(1/\kappa)} + \left(1 + \frac{\log 2}{\log(1/\kappa)}\right) \log\log\left(\frac{\lambda R L}{\delta^2}\right), \tag{37}$$

where $\phi(\beta) := \mathcal{L}_n(\beta) + \rho_\lambda(\beta)$ denotes the composite objective function. As clarified in the theorem statement, the squared tolerance δ^2 is not allowed to be arbitrarily small, which would contradict the fact that the composite gradient method may converge to a stationary point. However, our theory allows δ^2 to be of the same order as the squared *statistical error* $\epsilon_{\text{stat}}^2 = \|\widehat{\beta} - \beta^*\|_2^2$, the distance between a fixed global optimum and the target parameter β^*. From a statistical perspective, there is no point in optimizing beyond this tolerance.

With this setup, we now turn to a precise statement of our main optimization-theoretic result. As with Theorems 1 and 2, the statement of Theorem 3 is entirely deterministic.

Theorem 3 *Suppose the empirical loss $\mathcal{L}_n$ satisfies the RSC/RSM conditions* (34) *and* (35)*, and suppose the regularizer ρ_λ satisfies Assumption 1. Suppose $\widehat{\beta}$ is any global minimum of*

the program (30)*, with regularization parameters chosen such that*

$$\frac{8}{L} \cdot \max\left\{\|\nabla \mathcal{L}_n(\beta^*)\|_\infty,\ c'\tau\sqrt{\frac{\log p}{n}}\right\} \leq \lambda \leq \frac{c''\,\alpha}{RL}.$$

Suppose $\mu < 2\alpha$*. Then for any step size parameter* $\eta \geq \max\{2\alpha_3 - \mu,\, \mu\}$ *and tolerance* $\delta^2 \geq \frac{c\epsilon_{stat}^2}{1-\kappa} \cdot \frac{k \log p}{n}$*, we have*

$$\|\beta^t - \widehat{\beta}\|_2^2 \ \leq\ \frac{4}{2\alpha - \mu}\ \left(\delta^2 + \frac{\delta^4}{\tau} + c\tau\frac{k\log p}{n}\epsilon_{stat}^2\right), \qquad \forall t \geq T^*(\delta). \tag{38}$$

Remark: Note that for the optimal choice of tolerance parameter $\delta \asymp \frac{k\log p}{n}\epsilon_{\text{stat}}$, the error bound appearing in (38) takes the form $\frac{c\epsilon_{\text{stat}}^2}{2\alpha-\mu} \cdot \frac{k\log p}{n}$, meaning that successive iterates of the composite gradient descent algorithm are guaranteed to converge to a region within statistical accuracy of the true global optimum $\widehat{\beta}$. Concretely, if the sample size satisfies $n \gtrsim Ck\log p$ and the regularization parameters are chosen appropriately, Theorem 1 guarantees that $\epsilon_{\text{stat}} = \mathcal{O}\left(\sqrt{\frac{k\log p}{n}}\right)$ with high probability. Combined with Theorem 3, we then conclude that

$$\max\left\{\|\beta^t - \widehat{\beta}\|_2,\ \|\beta^t - \beta^*\|_2\right\} = \mathcal{O}\left(\sqrt{\frac{k\log p}{n}}\right),$$

for all iterations $t \geq T(\epsilon_{\text{stat}})$.

As would be expected, the (restricted) curvature α of the loss function and nonconvexity parameter μ of the penalty function enter into the bound via the denominator $2\alpha - \mu$. Indeed, the bound is tighter when the loss function possesses more curvature or the penalty function is closer to being convex, agreeing with intuition. Similar to our discussion in the remark following Theorem 2, the requirement $\mu < 2\alpha$ is certainly necessary for our proof technique, but it is possible that composite gradient descent still produces good results when this condition is violated. See Section 5 for simulations in scenarios involving mild and severe violations of this condition.

Finally, note that the parameter η must be sufficiently large (or equivalently, the step size must be sufficiently small) in order for the composite gradient descent algorithm to be well-behaved. See Nesterov (2007) for a discussion of how the step size may be chosen via an iterative search when the problem parameters are unknown.

In the case of corrected linear regression (Corollary 1), Lemma 13 of Loh and Wainwright (2012) establishes the RSC/RSM conditions for various statistical models. The following proposition shows that the conditions (34) and (35) hold in GLMs when the x_i's are drawn i.i.d. from a zero-mean sub-Gaussian distribution with parameter σ_x^2 and covariance matrix $\Sigma = \text{cov}(x_i)$. As usual, we assume a sample size $n \geq c\,k\log p$, for a sufficiently large constant $c > 0$. Recall the definition of the Taylor error $\mathcal{T}(\beta_1, \beta_2)$ from (32).

Proposition 1 *[RSC/RSM conditions for generalized linear models] There exists a constant* $\alpha_\ell > 0$*, depending only on the GLM and the parameters* (σ_x^2, Σ)*, such that for all*

vectors $\beta_2 \in \mathbb{B}_2(3) \cap \mathbb{B}_1(R)$*, we have*

$$\mathcal{T}(\beta_1, \beta_2) \geq \begin{cases} \frac{\alpha_\ell}{2}\|\Delta\|_2^2 - \frac{c^2\sigma_x^2}{2\alpha_\ell}\frac{\log p}{n}\|\Delta\|_1^2, & \text{for all } \|\beta_1 - \beta_2\|_2 \leq 3, \quad (39a) \\ \frac{3\alpha_\ell}{2}\|\Delta\|_2 - 3c\sigma_x\sqrt{\frac{\log p}{n}}\|\Delta\|_1, & \text{for all } \|\beta_1 - \beta_2\|_2 \geq 3, \quad (39b) \end{cases}$$

with probability at least $1 - c_1\exp(-c_2 n)$*. With the bound* $\|\psi''\|_\infty \leq \alpha_u$*, we also have*

$$\mathcal{T}(\beta_1, \beta_2) \leq \alpha_u \lambda_{max}(\Sigma) \left(\frac{3}{2}\|\Delta\|_2^2 + \frac{\log p}{n}\|\Delta\|_1^2\right), \qquad \text{for all } \beta_1, \beta_2 \in \mathbb{R}^p, \quad (40)$$

with probability at least $1 - c_1\exp(-c_2 n)$*.*

For the proof of Proposition 1, see Appendix D.

4.2 Form of Updates

In this section, we discuss how the updates (31) are readily computable in many cases. We begin with the case $\Omega = \mathbb{R}^p$, so we have no additional constraints apart from $g_{\lambda,\mu}(\beta) \leq R$. In this case, given iterate β^t, the next iterate β^{t+1} may be obtained via the following three-step procedure:

(1) First optimize the unconstrained program

$$\widehat{\beta} \in \arg\min_{\beta \in \mathbb{R}^p} \left\{ \frac{1}{2}\left\|\beta - \left(\beta^t - \frac{\nabla \bar{\mathcal{L}}_n(\beta^t)}{\eta}\right)\right\|_2^2 + \frac{\lambda}{\eta} \cdot g_{\lambda,\mu}(\beta) \right\}. \quad (41)$$

(2) If $g_{\lambda,\mu}(\widehat{\beta}) \leq R$, define $\beta^{t+1} = \widehat{\beta}$.

(3) Otherwise, if $g_{\lambda,\mu}(\widehat{\beta}) > R$, optimize the constrained program

$$\beta^{t+1} \in \arg\min_{g_{\lambda,\mu}(\beta) \leq R} \left\{ \frac{1}{2}\left\|\beta - \left(\beta^t - \frac{\nabla \bar{\mathcal{L}}_n(\beta^t)}{\eta}\right)\right\|_2^2 \right\}. \quad (42)$$

We derive the correctness of this procedure in Appendix C.1. For many nonconvex regularizers ρ_λ of interest, the unconstrained program (41) has a convenient closed-form solution: For the SCAD penalty (2), the program (41) has simple closed-form solution given by

$$\widehat{\beta}_{\text{SCAD}} = \begin{cases} 0 & \text{if } 0 \leq |z| \leq \nu\lambda, \\ z - \text{sign}(z) \cdot \nu\lambda & \text{if } \nu\lambda \leq |z| \leq (\nu+1)\lambda, \\ \frac{z - \text{sign}(z)\cdot\frac{a\nu\lambda}{a-1}}{1 - \frac{\nu}{a-1}} & \text{if } (\nu+1)\lambda \leq |z| \leq a\lambda, \\ z & \text{if } |z| \geq a\lambda. \end{cases} \quad (43)$$

For the MCP (3), the optimum of the program (41) takes the form

$$\widehat{\beta}_{\mathrm{MCP}} = \begin{cases} 0 & \text{if } 0 \leq |z| \leq \nu\lambda, \\ \frac{z - \mathrm{sign}(z) \cdot \nu\lambda}{1 - \nu/b} & \text{if } \nu\lambda \leq |z| \leq b\lambda, \\ z & \text{if } |z| \geq b\lambda. \end{cases} \tag{44}$$

In both (43) and (44), we have

$$z := \frac{1}{1 + \mu/\eta} \left(\beta^t - \frac{\nabla \overline{\mathcal{L}}_n(\beta^t)}{\eta} \right), \qquad \text{and} \qquad \nu := \frac{1/\eta}{1 + \mu/\eta},$$

and the operations are taken componentwise. See Appendix C.2 for the derivation of these closed-form updates.

More generally, when $\Omega \subsetneq \mathbb{R}^p$ (such as in the case of the graphical Lasso), the minimum in the program (31) must be taken over Ω, as well. Although the updates are not as simply stated, they still involve solving a convex optimization problem. Despite this more complicated form, however, our results from Section 4.1 on fast global convergence under restricted strong convexity and restricted smoothness assumptions carry over without modification, since they only require RSC/RSM conditions holding over a sufficiently small radius together with feasibility of β^*.

4.3 Proof of Theorem 3

We provide the outline of the proof here, with more technical results deferred to Appendix C. In broad terms, our proof is inspired by a result of Agarwal et al. (2012), but requires various modifications in order to be applied to the much larger family of nonconvex regularizers considered here.

Our first lemma shows that the optimization error $\beta^t - \widehat{\beta}$ lies in an approximate cone set:

Lemma 1 *Under the conditions of Theorem 3, suppose there exists a pair $(\bar{\eta}, T)$ such that*

$$\phi(\beta^t) - \phi(\widehat{\beta}) \leq \bar{\eta}, \qquad \forall t \geq T. \tag{45}$$

Then for any iteration $t \geq T$, we have

$$\|\beta^t - \widehat{\beta}\|_1 \leq 8\sqrt{k}\|\beta^t - \widehat{\beta}\|_2 + 16\sqrt{k}\|\widehat{\beta} - \beta^*\|_2 + 2 \cdot \min\left(\frac{2\bar{\eta}}{\lambda L}, R \right).$$

Our second lemma shows that as long as the composite gradient descent algorithm is initialized with a solution β^0 within a constant radius of a global optimum $\widehat{\beta}$, all successive iterates also lie within the same ball:

Lemma 2 *Under the conditions of Theorem 3, and with an initial vector β^0 such that $\|\beta^0 - \widehat{\beta}\|_2 \leq 3$, we have*

$$\|\beta^t - \widehat{\beta}\|_2 \leq 3, \qquad \textit{for all } t \geq 0. \tag{46}$$

In particular, suppose we initialize the composite gradient procedure with a vector β^0 such that $\|\beta^0\|_2 \leq \frac{3}{2}$. Then by the triangle inequality,

$$\|\beta^0 - \widehat{\beta}\|_2 \leq \|\beta^0\|_2 + \|\widehat{\beta} - \beta^*\|_2 + \|\beta^*\|_2 \leq 3,$$

where we have assumed our scaling of n guarantees $\|\widehat{\beta} - \beta^*\|_2 \leq 1/2$.

Finally, recalling our earlier definition (36) of κ, the third lemma combines the results of Lemmas 1 and 2 to establish a bound on the value of the objective function that decays exponentially with t:

Lemma 3 *Under the same conditions of Lemma 2, suppose in addition that* (45) *holds and* $\frac{32k\tau \log p}{n} \leq \frac{2\alpha - \mu}{4}$. *Then for any* $t \geq T$, *we have*

$$\phi(\beta^t) - \phi(\widehat{\beta}) \leq \kappa^{t-T}(\phi(\beta^T) - \phi(\widehat{\beta})) + \frac{\xi}{1-\kappa}(\epsilon^2 + \bar{\epsilon}^2),$$

where $\bar{\epsilon} := 8\sqrt{k}\epsilon_{stat}$, $\epsilon := 2 \cdot \min\left(\frac{2\bar{\eta}}{\lambda L}, R\right)$, *the quantities* κ *and* φ *are defined according to* (36), *and*

$$\xi := \frac{1}{1 - \varphi(n,p,k)} \cdot \frac{\tau \log p}{n} \cdot \left(\frac{2\alpha - \mu}{4\eta} + 2\varphi(n,p,k) + 5\right). \tag{47}$$

The remainder of the proof follows an argument used in Agarwal et al. (2012), so we only provide a high-level sketch. We first prove the following inequality:

$$\phi(\beta^t) - \phi(\widehat{\beta}) \leq \delta^2, \qquad \text{for all } t \geq T^*(\delta), \tag{48}$$

as follows. We divide the iterations $t \geq 0$ into a series of epochs $[T_\ell, T_{\ell+1})$ and define tolerances $\bar{\eta}_0 > \bar{\eta}_1 > \cdots$ such that

$$\phi(\beta^t) - \phi(\widehat{\beta}) \leq \bar{\eta}_\ell, \qquad \forall t \geq T_\ell.$$

In the first iteration, we apply Lemma 3 with $\bar{\eta}_0 = \phi(\beta^0) - \phi(\widehat{\beta})$ to obtain

$$\phi(\beta^t) - \phi(\widehat{\beta}) \leq \kappa^t \left(\phi(\beta^0) - \phi(\widehat{\beta})\right) + \frac{\xi}{1-\kappa}(4R^2 + \bar{\epsilon}^2), \qquad \forall t \geq 0.$$

Let $\bar{\eta}_1 := \frac{2\xi}{1-\kappa}(4R^2 + \bar{\epsilon}^2)$, and note that for $T_1 := \left\lceil \frac{\log(2\bar{\eta}_0/\bar{\eta}_1)}{\log(1/\kappa)} \right\rceil$, we have

$$\phi(\beta^t) - \phi(\widehat{\beta}) \leq \bar{\eta}_1 \leq \frac{4\xi}{1-\kappa} \max\{4R^2, \bar{\epsilon}^2\}, \qquad \text{for all } t \geq T_1.$$

For $\ell \geq 1$, we now define

$$\bar{\eta}_{\ell+1} := \frac{2\xi}{1-\kappa}(\epsilon_\ell^2 + \bar{\epsilon}^2), \quad \text{and} \quad T_{\ell+1} := \left\lceil \frac{\log(2\bar{\eta}_\ell/\bar{\eta}_{\ell+1})}{\log(1/\kappa)} \right\rceil + T_\ell,$$

where $\epsilon_\ell := 2\min\left\{\frac{\bar{\eta}_\ell}{\lambda L}, R\right\}$. From Lemma 3, we have

$$\phi(\beta^t) - \phi(\widehat{\beta}) \leq \kappa^{t-T_\ell}\left(\phi(\beta^{T_\ell}) - \phi(\widehat{\beta})\right) + \frac{\xi}{1-\kappa}(\epsilon_\ell^2 + \bar{\epsilon}^2), \qquad \text{for all } t \geq T_\ell,$$

implying by our choice of $\{(\eta_\ell, T_\ell)\}_{\ell \geq 1}$ that

$$\phi(\beta^t) - \phi(\widehat{\beta}) \leq \bar{\eta}_{\ell+1} \leq \frac{4\xi}{1-\kappa}\max\{\epsilon_\ell^2, \bar{\epsilon}^2\}, \qquad \forall t \geq T_{\ell+1}.$$

Finally, we use the recursion

$$\bar{\eta}_{\ell+1} \leq \frac{4\xi}{1-\kappa}\max\{\epsilon_\ell^2, \bar{\epsilon}^2\}, \qquad T_\ell \leq \ell + \frac{\log(2^\ell \bar{\eta}_0/\bar{\eta}_\ell)}{\log(1/\kappa)}, \tag{49}$$

to establish the recursion

$$\bar{\eta}_{\ell+1} \leq \frac{\bar{\eta}_\ell}{4^{2^{\ell-1}}}, \qquad \frac{\bar{\eta}_{\ell+1}}{\lambda L} \leq \frac{R}{4^{2^\ell}}. \tag{50}$$

Inequality (48) then follows from computing the number of epochs and time steps necessary to obtain $\frac{\lambda R L}{4^{2^{\ell-1}}} \leq \delta^2$. For the remaining steps used to obtain (50) from (49), we refer the reader to Agarwal et al. (2012).

Finally, by (85b) in the proof of Lemma 3 in Appendix C.5 and the relative scaling of (n, p, k), we have

$$\begin{aligned}\frac{2\alpha - \mu}{4}\|\beta^t - \widehat{\beta}\|_2^2 &\leq \phi(\beta^t) - \phi(\widehat{\beta}) + 2\tau\frac{\log p}{n}\left(\frac{2\delta^2}{\lambda L} + \bar{\epsilon}\right)^2 \\ &\leq \delta^2 + 2\tau\frac{\log p}{n}\left(\frac{2\delta^2}{\lambda L} + \bar{\epsilon}\right)^2,\end{aligned}$$

where we have set $\epsilon = \frac{2\delta^2}{\lambda L}$. Rearranging and performing some algebra with our choice of λ gives the ℓ_2-bound.

5. Simulations

In this section, we report the results of simulations we performed to validate our theoretical results. In particular, we present results for two versions of the loss function $\mathcal{L}_n$, corresponding to linear and logistic regression, and three penalty functions, namely the ℓ_1-norm (Lasso), the SCAD penalty, and the MCP, as detailed in Section 2.2. In all cases, we chose regularization parameters $R = \frac{1.1}{\lambda} \cdot \rho_\lambda(\beta^*)$, to ensure feasibility of β^*, and $\lambda = \sqrt{\frac{\log p}{n}}$; in practical applications where β^* is unknown, we would need to tune λ and R using a method such as cross-validation.

Linear regression: In the case of linear regression, we simulated covariates corrupted by additive noise according to the mechanism described in Section 3.2, giving the estimator

$$\widehat{\beta} \in \arg\min_{g_{\lambda,\mu}(\beta) \leq R}\left\{\frac{1}{2}\beta^T\left(\frac{Z^T Z}{n} - \Sigma_w\right)\beta - \frac{y^T Z}{n}\beta + \rho_\lambda(\beta)\right\}. \tag{51}$$

We generated i.i.d. samples $x_i \sim N(0, I)$ and set $\Sigma_w = (0.2)^2 I$, and generated additive noise $\epsilon_i \sim N(0, (0.1)^2)$.

Logistic regression: In the case of logistic regression, we also generated i.i.d. samples $x_i \sim N(0, I)$. Since $\psi(t) = \log(1 + \exp(t))$, the program (15) becomes

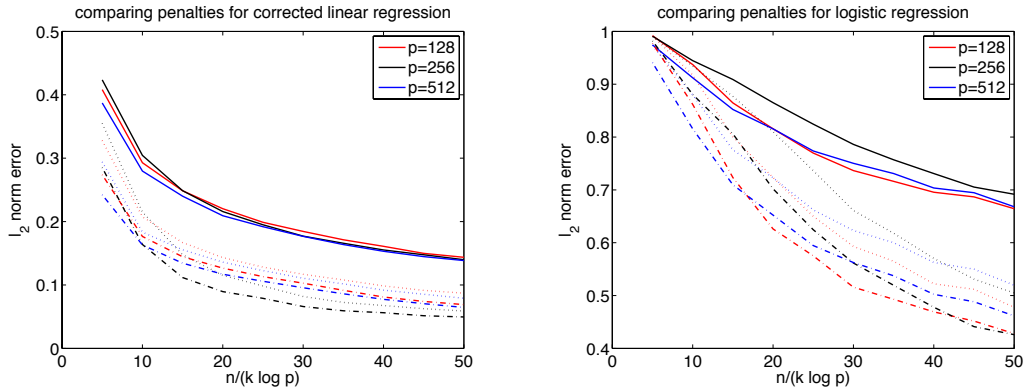
$$\widehat{\beta} \in \arg \min_{g_{\lambda,\mu}(\beta) \leq R} \left\{ \frac{1}{n} \sum_{i=1}^{n} \{\log(1 + \exp(\langle \beta, x_i \rangle) - y_i \langle \beta, x_i \rangle\} + \rho_\lambda(\beta) \right\}. \tag{52}$$

We optimized the programs (51) and (52) using the composite gradient updates (31). In order to compute the updates, we used the three-step procedure described in Section 4.2, together with the updates for SCAD and MCP given by (43) and (44). Note that the updates for the Lasso penalty may be generated more simply and efficiently as discussed in Agarwal et al. (2012).

Figure 2 shows the results of corrected linear regression with Lasso, SCAD, and MCP regularizers for three different problem sizes p. In each case, β^* is a k-sparse vector with $k = \lfloor \sqrt{p} \rfloor$, where the nonzero entries were generated from a normal distribution and the vector was then rescaled so that $\|\beta^*\|_2 = 1$. As predicted by Theorem 1, the three curves corresponding to the same penalty function stack up when the estimation error $\|\widehat{\beta} - \beta^*\|_2$ is plotted against the rescaled sample size $\frac{n}{k \log p}$, and the ℓ_2-error decreases to zero as the number of samples increases, showing that the estimators (51) and (52) are statistically consistent. The Lasso, SCAD, and MCP regularizers are depicted by solid, dotted, and dashed lines, respectively. We chose the parameter $a = 3.7$ for the SCAD penalty, suggested by Fan and Li (2001) to be "optimal" based on cross-validated empirical studies, and chose $b = 3.5$ for the MCP. Each point represents an average over 20 trials.

The simulations in Figure 3 depict the optimization-theoretic conclusions of Theorem 3. Each panel shows two different families of curves, depicting the statistical error $\log(\|\widehat{\beta} - \beta^*\|_2)$ in red and the optimization error $\log(\|\beta^t - \widehat{\beta}\|_2)$ in blue. Here, the vertical axis measures the ℓ_2-error on a logarithmic scale, while the horizontal axis tracks the iteration number. Within each panel, the blue curves were obtained by running the composite gradient descent algorithm from 10 different initial starting points chosen at random, and the optimization error is measured with respect to a stationary point obtained from an earlier run of the composite gradient descent algorithm in place of $\widehat{\beta}$, since a global optimum is unknown. The statistical error is similarly displayed as the distance between β^* and the stationary points computed from successive runs of composite gradient descent. In all cases, we used the parameter settings $p = 128$, $k = \lfloor \sqrt{p} \rfloor$, and $n = \lfloor 20 k \log p \rfloor$. As predicted by our theory, the optimization error decreases at a linear rate (on the log scale) until it falls to the level of statistical error. Furthermore, it is interesting to compare the plots in panels (c) and (d), which provide simulation results for two different values of the SCAD parameter a. We see that the choice $a = 3.7$ leads to a tighter cluster of optimization trajectories, providing further evidence that this setting suggested by Fan and Li (2001) is in some sense optimal.

Figure 4 provides analogous results to Figure 3 in the case of logistic regression, using $p = 64, k = \lfloor \sqrt{p} \rfloor$, and $n = \lfloor 20 k \log p \rfloor$. The plot shows solution trajectories for 20 different



(a) (b)

Figure 2: Plots showing statistical consistency of linear and logistic regression with Lasso, SCAD, and MCP regularizers, and with sparsity level $k = \lfloor \sqrt{p} \rfloor$. Panel (a) shows results for corrected linear regression, where covariates are subject to additive noise with $SNR = 5$. Panel (b) shows similar results for logistic regression. Each point represents an average over 20 trials. In both cases, the estimation error $\|\widehat{\beta} - \beta^*\|_2$ is plotted against the rescaled sample size $\frac{n}{k \log p}$. Lasso, SCAD, and MCP results are represented by solid, dotted, and dashed lines, respectively. As predicted by Theorem 1 and Corollaries 1 and 2, the curves for each of the three types stack up for different problem sizes p, and the error decreases to zero as the number of samples increases, showing that our methods are statistically consistent.

initializations of composite gradient descent. Again, we see that the log optimization error decreases at a linear rate up to the level of statistical error, as predicted by Theorem 3. Furthermore, the Lasso penalty yields a unique global optimum $\widehat{\beta}$, since the program (52) is convex, as we observe in panel (a). In contrast, the nonconvex program based on the SCAD penalty produces multiple local optima, whereas the MCP yields a relatively large number of local optima. Note that empirically, all local optima appear to lie within the small ball around β^* defined in Theorem 1. However, if we use $\lambda_{\min}(\nabla^2 \mathcal{L}_n(\beta^*))$ as a surrogate for α_1, we see that $2\alpha_1 < \mu$ in the case of the SCAD or MCP regularizers, which is not covered by our theory.

Finally, Figure 5 explores the behavior of our algorithm when the condition $\mu < 2\alpha_1$ from Theorem 1 is significantly violated. We generated i.i.d. samples $x_i \sim N(0, \Sigma)$, with Σ taken to be a Toeplitz matrix with entries $\Sigma_{ij} = \zeta^{|i-j|}$, for some parameter $\zeta \in [0, 1)$, so that $\lambda_{\min}(\Sigma) \geq (1 - \zeta)^2$. We chose $\zeta \in \{0.5, 0.9\}$, resulting in $\alpha_1 \approx \{0.25, 0.01\}$. The problem parameters were chosen to be $p = 512, k = \lfloor \sqrt{p} \rfloor$, and $n = \lfloor 10k \log p \rfloor$. Panel (a) shows the expected good behavior of ℓ_1-regularization, even for $\alpha_1 = 0.01$; although convergence is slow and the overall statistical error is greater than for $\Sigma = I$ (cf. Figure 3(a)), composite gradient descent still converges at a linear rate. Panel (b) shows that for SCAD parameter $a = 2.5$ (corresponding to $\mu \approx 0.67$), local optima still seem to be well-behaved even for

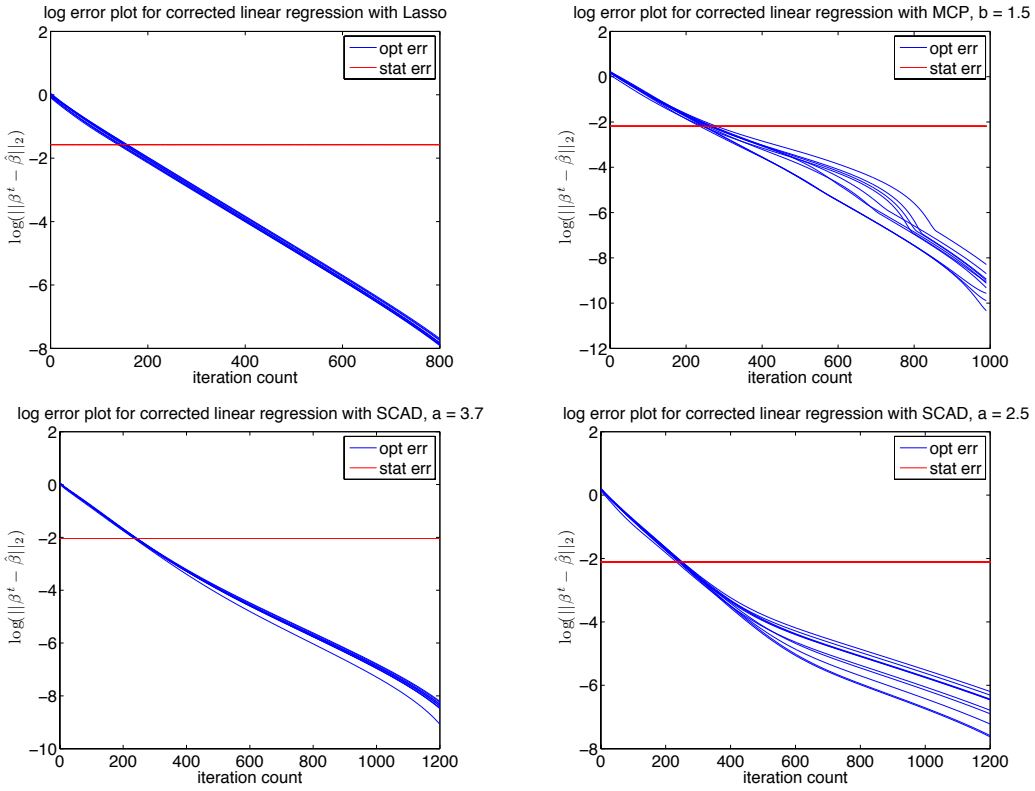


Figure 3: Plots illustrating linear rates of convergence on a log scale for corrected linear regression with Lasso, MCP, and SCAD regularizers, with $p = 128$, $k = \lfloor \sqrt{p} \rfloor$, and $n = \lfloor 20k \log p \rfloor$, where covariates are corrupted by additive noise with $SNR = 5$. Red lines depict statistical error $\log\left(\|\widehat{\beta} - \beta^*\|_2\right)$ and blue lines depict optimization error $\log\left(\|\beta^t - \widehat{\beta}\|_2\right)$. As predicted by Theorem 3, the optimization error decreases linearly when plotted against the iteration number on a log scale, up to statistical accuracy. Each plot shows the solution trajectory for 10 different initializations of the composite gradient descent algorithm. Panels (a) and (b) show the results for Lasso and MCP regularizers, respectively; panels (c) and (d) show results for the SCAD penalty with two different parameter values. Note that the empirically optimal choice $a = 3.7$ proposed by Fan and Li (2001) generates solution paths that exhibit a smaller spread than the solution paths generated for a smaller setting of the parameter a.

$2\alpha_1 = 0.5 < \mu$. However, for much smaller values of α_1, the good behavior breaks down, as seen in panels (c) and (d). Note that in the latter two panels, the composite gradient descent algorithm does not appear to be converging, even as the iteration number increases. Comparing (c) and (d) also illustrates the interplay between the curvature parameter α_1 of $\mathcal{L}_n$ and the nonconvexity parameter μ of ρ_λ. Indeed, the plot in panel (d) is slightly "better" than the plot in panel (c), in the sense that initial iterates at least demonstrate

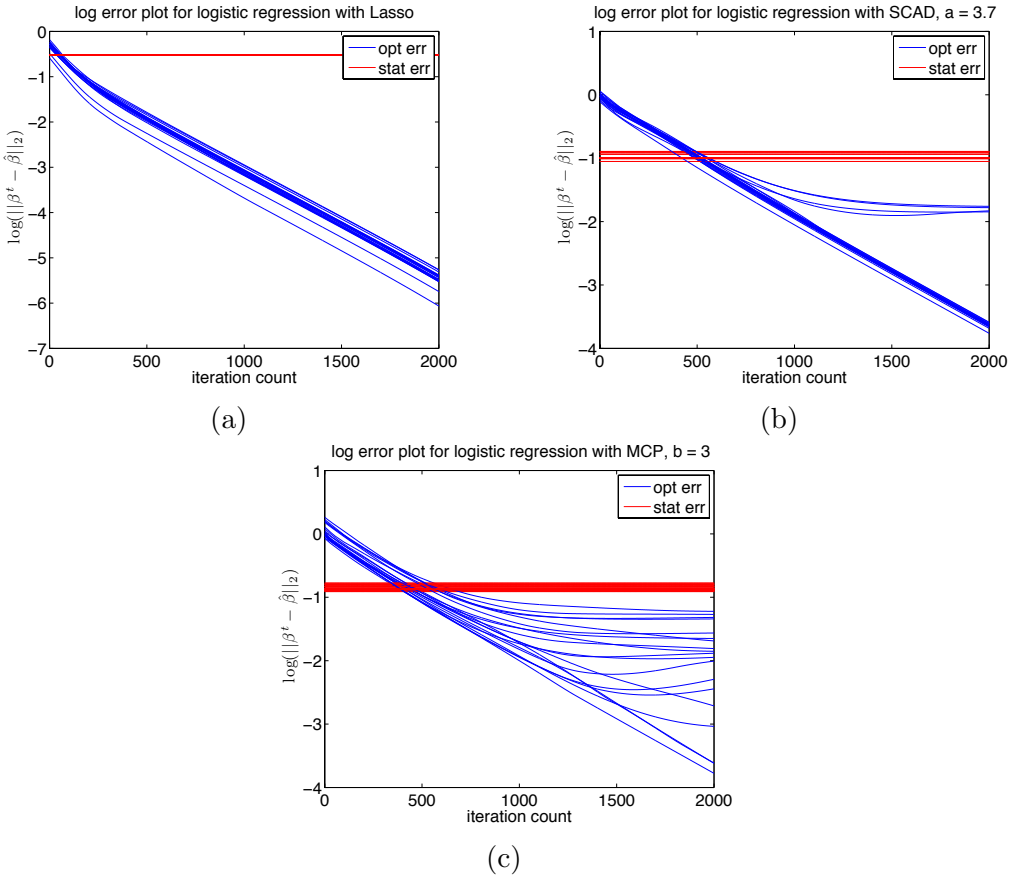


Figure 4: Plots that demonstrate linear rates of convergence on a log scale for logistic regression with $p = 64, k = \sqrt{p}$, and $n = \lfloor 20k \log p \rfloor$. Red lines depict statistical error $\log\left(\|\widehat{\beta} - \beta^*\|_2\right)$ and blue lines depict optimization error $\log\left(\|\beta^t - \widehat{\beta}\|_2\right)$. (a) Lasso penalty. (b) SCAD penalty. (c) MCP. As predicted by Theorem 3, the optimization error decreases linearly when plotted against the iteration number on a log scale, up to statistical accuracy. Each plot shows the solution trajectory for 20 different initializations of the composite gradient descent algorithm. Multiple local optima emerge in panels (b) and (c), due to nonconvex regularizers.

some pattern of convergence. This could be attributed to the fact that the SCAD parameter is larger, corresponding to a smaller value of μ.

6. Discussion

We have analyzed theoretical properties of local optima of regularized M-estimators, where both the loss and penalty function are allowed to be nonconvex. Our results are the first to establish that *all stationary points* of such nonconvex problems are close to the truth, implying that any optimization method guaranteed to converge to a stationary point will

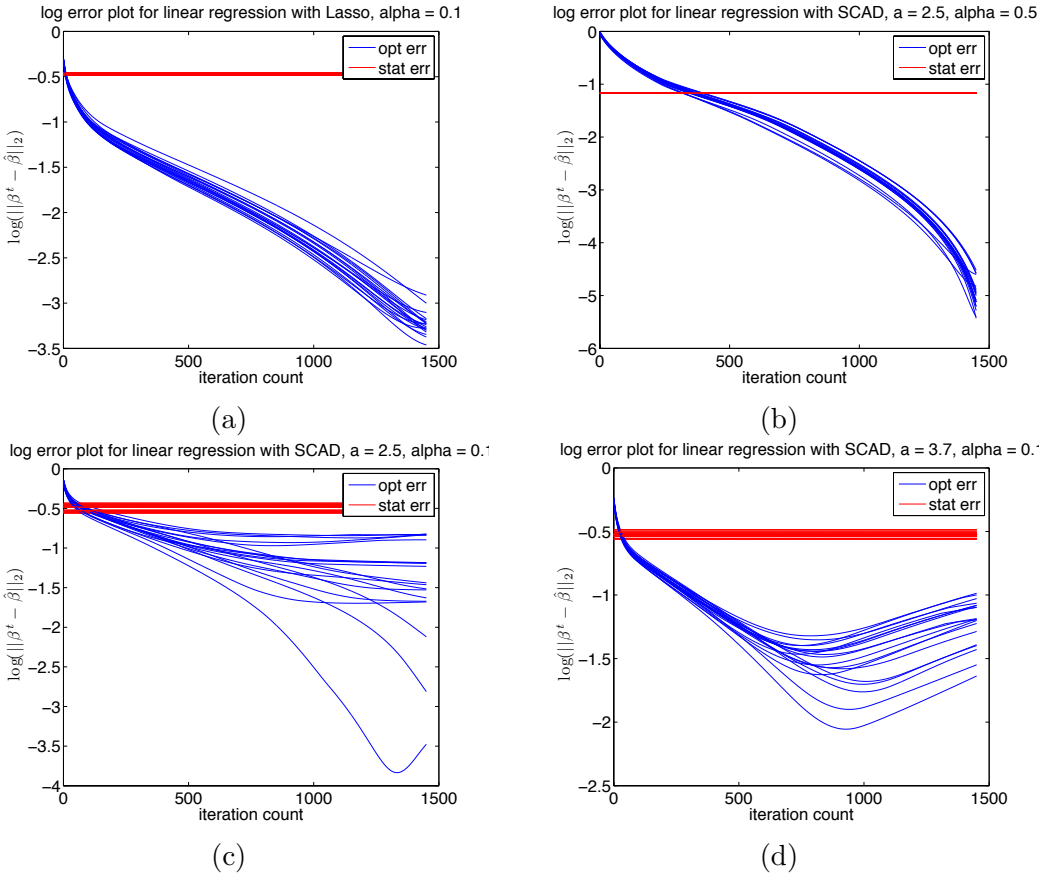


Figure 5: Plots showing breakdown points as a function of the curvature parameter α_1 of the loss function and the nonconvexity parameter μ of the penalty function. The loss comes from ordinary least squares linear regression, where covariates are fully-observed and sampled from a Gaussian distribution with covariance equal to a Toeplitz matrix. Panel (a) depicts the good behavior of Lasso-based linear regression. Panel (b) shows that local optima may still be well-behaved even when $2\alpha_1 < \mu$, although this situation is not covered by our theory. Panels (c) and (d) show that the good behavior nonetheless disintegrates for very small values of α_1 when the regularizer is nonconvex.

provide statistically consistent solutions. We show concretely that a variant of composite gradient descent may be used to obtain near-global optima in linear time, and verify our theoretical results with simulations.

Future directions of research include further generalizing our statistical consistency results to other nonconvex regularizers not covered by our present theory, such as bridge penalties or regularizers that do not decompose across coordinates. In addition, it would be interesting to expand our theory to nonsmooth loss functions such as the hinge loss. For both nonsmooth losses and nonsmooth penalties (including capped-ℓ_1), it remains an open

question whether a modified version of composite gradient descent may be used to obtain near-global optima in polynomial time. Finally, it would be useful to develop a general method for establishing RSC and RSM conditions, beyond the specialized methods used for studying GLMs in this paper.

Acknowledgments

The work of PL was supported from a Hertz Foundation Fellowship and an NSF Graduate Research Fellowship. In addition, PL acknowledges support from a PD Award, and MJW from a DY Award. MJW and PL were also partially supported by NSF grant CIF-31712-23800. The authors thank the associate editors and anonymous reviewers for their helpful suggestions that improved the manuscript.

Appendix A. Properties of Regularizers

In this section, we establish properties of some nonconvex regularizers covered by our theory (Appendix A.1) and verify that specific regularizers satisfy Assumption 1 (Appendix A.2). The properties given in Appendix A.1 are used in the proof of Theorem 1.

A.1 General Properties

We begin with some general properties of regularizers that satisfy Assumption 1.

Lemma 4

(a) Under conditions (i)–(ii) of Assumption 1, conditions (iii) and (iv) together imply that ρ_λ is λL-Lipschitz as a function of t. In particular, all subgradients and derivatives of ρ_λ are bounded in magnitude by λL.

(b) Under the conditions of Assumption 1, we have

$$\lambda L \|\beta\|_1 \leq \rho_\lambda(\beta) + \frac{\mu}{2}\|\beta\|_2^2, \qquad \forall \beta \in \mathbb{R}^p. \tag{53}$$

Proof (a): Suppose $0 \leq t_1 \leq t_2$. Then

$$\frac{\rho_\lambda(t_2) - \rho_\lambda(t_1)}{t_2 - t_1} \leq \frac{\rho_\lambda(t_1)}{t_1},$$

by condition (iii). Applying (iii) once more, we have

$$\frac{\rho_\lambda(t_1)}{t_1} \leq \lim_{t \to 0^+} \frac{\rho_\lambda(t)}{t} = \lambda L,$$

where the last equality comes from condition (iv). Hence,

$$0 \leq \rho_\lambda(t_2) - \rho_\lambda(t_1) \leq \lambda L(t_2 - t_1).$$

A similar argument applies to the cases when one (or both) of t_1 and t_2 are negative.

(b): Clearly, it suffices to verify the inequality for the scalar case:

$$\lambda L t \leq \rho_\lambda(t) + \frac{\mu t^2}{2}, \qquad \forall t \in \mathbb{R}.$$

The inequality is trivial for $t = 0$. For $t > 0$, the convexity of the right-hand expression implies that for any $s \in (0, t)$, we have

$$\left(\rho_\lambda(t) + \frac{\mu t^2}{2}\right) - \left(\rho_\lambda(0) + \frac{\mu \cdot 0^2}{2}\right) \geq (t - 0) \cdot \left(\rho'_\lambda(s) + \mu s\right).$$

Taking a limit as $s \to 0^+$ then yields the desired inequality. The case $t < 0$ follows by symmetry. ■

Lemma 5 *Suppose ρ_λ satisfies the conditions of Assumption 1. Let $v \in \mathbb{R}^p$, and let A denote the index set of the k largest elements of v in magnitude. Suppose $\xi > 0$ is such that $\xi\rho_\lambda(v_A) - \rho_\lambda(v_{A^c}) \geq 0$. Then*

$$\xi\rho_\lambda(v_A) - \rho_\lambda(v_{A^c}) \leq \lambda L(\xi\|v_A\|_1 - \|v_{A^c}\|_1). \tag{54}$$

Moreover, if $\beta^* \in \mathbb{R}^p$ is k-sparse, then for an vector $\beta \in \mathbb{R}^p$ such that $\xi\rho_\lambda(\beta^*) - \rho_\lambda(\beta) > 0$ and $\xi \geq 1$, we have

$$\xi\rho_\lambda(\beta^*) - \rho_\lambda(\beta) \leq \lambda L\big(\xi\|\nu_A\|_1 - \|\nu_{A^c}\|_1\big), \tag{55}$$

where $\nu := \beta - \beta^*$ and A is the index set of the k largest elements of ν in magnitude.

Proof We first establish (54). Define $f(t) := \frac{t}{\rho_\lambda(t)}$ for $t > 0$. By our assumptions on ρ_λ, the function f is nondecreasing in $|t|$, so

$$\|v_{A^c}\|_1 = \sum_{j \in A^c} \rho_\lambda(v_j) \cdot f(|v_j|) \leq \sum_{j \in A^c} \rho_\lambda(v_j) \cdot f(\|v_{A^c}\|_\infty) = \rho_\lambda(v_{A^c}) \cdot f\left(\|v_{A^c}\|_\infty\right). \tag{56}$$

Again using the nondecreasing property of f, we have

$$\rho_\lambda(v_A) \cdot f(\|v_{A^c}\|_\infty) = \sum_{j \in A} \rho_\lambda(v_j) \cdot f(\|v_{A^c}\|_\infty) \leq \sum_{j \in A} \rho_\lambda(v_j) \cdot f(|v_j|) = \|v_A\|_1. \tag{57}$$

Note that for $t > 0$, we have

$$f(t) \geq \lim_{s \to 0^+} f(s) = \lim_{s \to 0^+} \frac{s - 0}{\rho_\lambda(s) - \rho_\lambda(0)} = \frac{1}{\lambda L},$$

where the last equality follows from condition (iv) of Assumption 1. Combining this result with (56) and (57) yields

$$0 \leq \xi\rho_\lambda(v_A) - \rho_\lambda(v_{A^c}) \leq \frac{1}{f(\|v_{A^c}\|_\infty)} \cdot \big(\xi\|v_A\|_1 - \|v_{A^c}\|_1\big) \leq \lambda L\big(\xi\|v_A\|_1 - \|v_{A^c}\|_1\big),$$

as claimed.

We now turn to the proof of the bound (55). Letting $S := \text{supp}(\beta^*)$ denote the support of β^*, the triangle inequality and subadditivity of ρ (see the remark following Assumption 1; cf. Lemma 1 of Chen and Gu, 2014) imply that

$$\begin{aligned} 0 \leq \xi\rho_\lambda(\beta^*) - \rho_\lambda(\beta) &= \xi\rho_\lambda(\beta^*_S) - \rho_\lambda(\beta_S) - \rho_\lambda(\beta_{S^c}) \\ &\leq \xi\rho_\lambda(\nu_S) - \rho_\lambda(\beta_{S^c}) \\ &= \xi\rho_\lambda(\nu_S) - \rho_\lambda(\nu_{S^c}) \\ &\leq \xi\rho_\lambda(\nu_A) - \rho_\lambda(\nu_{A^c}) \\ &\leq \lambda L\big(\xi\|\nu_A\|_1 - \|\nu_{A^c}\|_1\big), \end{aligned}$$

thereby completing the proof. ■

A.2 Verification for Specific Regularizers

We now verify that Assumption 1 is satisfied by the SCAD and MCP regularizers. (The properties are trivial to verify for the Lasso penalty.)

Lemma 6 *The SCAD regularizer* (2) *with parameter a satisfies the conditions of Assumption 1 with* $L = 1$ *and* $\mu = \frac{1}{a-1}$.

Proof Conditions (i)–(iii) were already verified in Zhang and Zhang (2012). Furthermore, we may easily compute the derivative of the SCAD regularizer to be

$$\frac{\partial}{\partial t}\rho_\lambda(t) = \text{sign}(t) \cdot \left(\lambda \cdot \mathbb{I}\left\{|t| \leq \lambda\right\} + \frac{(a\lambda - |t|)_+}{a-1} \cdot \mathbb{I}\left\{|t| > \lambda\right\}\right), \qquad t \neq 0, \tag{58}$$

and any point in the interval $[-\lambda, \lambda]$ is a valid subgradient at $t = 0$, so condition (iv) is satisfied for any $L \geq 1$. Furthermore, we have $\frac{\partial^2}{\partial t^2}\rho_\lambda(t) \geq \frac{-1}{a-1}$, so $\rho_{\lambda,\mu}$ is convex whenever $\mu \geq \frac{1}{a-1}$, giving condition (v). ■

Lemma 7 *The MCP regularizer* (3) *with parameter b satisfies the conditions of Assumption 1 with1* $L = 1$ *and* $\mu = \frac{1}{b}$.

Proof Again, the conditions (i)–(iii) are already verified in Zhang and Zhang (2012). We may compute the derivative of the MCP regularizer to be

$$\frac{\partial}{\partial t}\rho_\lambda(t) = \lambda \cdot \text{sign}(t) \cdot \left(1 - \frac{|t|}{\lambda b}\right)_+, \qquad t \neq 0, \tag{59}$$

with subgradient $\lambda[-1, +1]$ at $t = 0$, so condition (iv) is again satisfied for any $L \geq 1$. Taking another derivative, we have $\frac{\partial^2}{\partial t^2}\rho_\lambda(t) \geq \frac{-1}{b}$, so condition (v) of Assumption 1 holds with $\mu = \frac{1}{b}$. ■

Appendix B. Proofs of Corollaries in Section 3

In this section, we provide proofs of the corollaries to Theorem 1 stated in Section 3. Throughout this section, we use the convenient shorthand notation

$$\mathcal{E}_n(\Delta) := \langle \nabla \mathcal{L}_n(\beta^* + \Delta) - \nabla \mathcal{L}_n(\beta^*), \Delta \rangle. \tag{60}$$

B.1 General Results for Verifying RSC

We begin with two lemmas that will be useful for establishing the RSC conditions (4) in the special case where $\mathcal{L}_n$ is convex. We assume throughout that $\|\Delta\|_1 \leq 2R$, since β^* and $\beta^* + \Delta$ lie in the feasible set.

Lemma 8 *Suppose $\mathcal{L}_n$ is convex. If condition* (4a) *holds and $n \geq 4R^2\tau_1^2 \log p$, then*

$$\mathcal{E}_n(\Delta) \geq \alpha_1 \|\Delta\|_2 - \sqrt{\frac{\log p}{n}} \|\Delta\|_1, \qquad \text{for all } \|\Delta\|_2 \geq 1. \tag{61}$$

Proof Fix an arbitrary $\Delta \in \mathbb{R}^p$ with $\|\Delta\|_2 \geq 1$. Since $\mathcal{L}_n$ is convex, the function $f : [0,1] \to \mathbb{R}$ given by $f(t) := \mathcal{L}_n(\beta^* + t\Delta)$ is also convex, so $f'(1) - f'(0) \geq f'(t) - f'(0)$ for all $t \in [0,1]$. Computing the derivatives of f yields the inequality

$$\mathcal{E}_n(\Delta) \;=\; \langle \nabla \mathcal{L}_n(\beta^* + \Delta) - \nabla \mathcal{L}_n(\beta^*), \Delta \rangle \geq \frac{1}{t} \langle \nabla \mathcal{L}_n(\beta^* + t\Delta) - \nabla \mathcal{L}_n(\beta^*), t\Delta \rangle.$$

Taking $t = \frac{1}{\|\Delta\|_2} \in (0,1]$ and applying condition (4a) to the rescaled vector $\frac{\Delta}{\|\Delta\|_2}$ then yields

$$\begin{aligned}
\mathcal{E}_n(\Delta) &\geq \|\Delta\|_2 \left(\alpha_1 - \tau_1 \frac{\log p}{n} \frac{\|\Delta\|_1^2}{\|\Delta\|_2^2} \right) \\
&\geq \|\Delta\|_2 \left(\alpha_1 - \frac{2R\tau_1 \log p}{n} \frac{\|\Delta\|_1}{\|\Delta\|_2^2} \right) \\
&\geq \|\Delta\|_2 \left(\alpha_1 - \sqrt{\frac{\log p}{n}} \frac{\|\Delta\|_1}{\|\Delta\|_2} \right) \\
&= \alpha_1 \|\Delta\|_2 - \sqrt{\frac{\log p}{n}} \|\Delta\|_1,
\end{aligned}$$

where the third inequality uses the assumption on the relative scaling of (n,p) and the fact that $\|\Delta\|_2 \geq 1$. ■

On the other hand, if (4a) holds globally over $\Delta \in \mathbb{R}^p$, we obtain (4b) for free:

Lemma 9 *If inequality* (4a) *holds for all $\Delta \in \mathbb{R}^p$ and $n \geq 4R^2\tau_1^2 \log p$, then* (4b) *holds, as well.*

Proof Suppose $\|\Delta\|_2 \geq 1$. Then

$$\alpha_1 \|\Delta\|_2^2 - \tau_1 \frac{\log p}{n} \|\Delta\|_1^2 \geq \alpha_1 \|\Delta\|_2 - 2R\tau_1 \frac{\log p}{n} \|\Delta\|_1 \geq \alpha_1 \|\Delta\|_2 - \sqrt{\frac{\log p}{n}} \|\Delta\|_1,$$

again using the assumption on the scaling of (n,p). ■

B.2 Proof of Corollary 1

Note that $\mathcal{E}_n(\Delta) = \Delta^T \widehat{\Gamma} \Delta$, so in particular,

$$\mathcal{E}_n(\Delta) \geq \Delta^T \Sigma_x \Delta - |\Delta^T (\Sigma_x - \widehat{\Gamma}) \Delta|.$$

Applying Lemma 12 in Loh and Wainwright (2012) with $s = \frac{n}{\log p}$ to bound the second term, we have

$$\begin{aligned}
\mathcal{E}_n(\Delta) &\geq \lambda_{\min}(\Sigma_x) \|\Delta\|_2^2 - \left(\frac{\lambda_{\min}(\Sigma_x)}{2} \|\Delta\|_2^2 + \frac{c \log p}{n} \|\Delta\|_1^2 \right) \\
&= \frac{\lambda_{\min}(\Sigma_x)}{2} \|\Delta\|_2^2 - \frac{c \log p}{n} \|\Delta\|_1^2,
\end{aligned}$$

a bound which holds for all $\Delta \in \mathbb{R}^p$ with probability at least $1 - c_1 \exp(-c_2 n)$ whenever $n \gtrsim k \log p$. Then Lemma 9 in Appendix B.1 implies that the RSC condition (4b) holds. It remains to verify the validity of the specified choice of λ. We have

$$\begin{aligned}
\|\nabla \mathcal{L}_n(\beta^*)\|_\infty = \|\widehat{\Gamma} \beta^* - \widehat{\gamma}\|_\infty &= \|(\widehat{\gamma} - \Sigma_x \beta^*) + (\Sigma_x - \widehat{\Gamma}) \beta^*\|_\infty \\
&\leq \|(\widehat{\gamma} - \Sigma_x \beta^*)\|_\infty + \|(\Sigma_x - \widehat{\Gamma}) \beta^*\|_\infty.
\end{aligned}$$

As shown in previous work (Loh and Wainwright, 2012), both of these terms are upper-bounded by $c' \varphi \sqrt{\frac{\log p}{n}}$ with high probability. Consequently, the claim in the corollary follows by applying Theorem 1.

B.3 Proof of Corollary 2

In the case of GLMs, we have

$$\mathcal{E}_n(\Delta) = \frac{1}{n} \sum_{i=1}^n (\psi'(\langle x_i, \, \beta^* + \Delta \rangle) - \psi'(\langle x_i, \, \beta^* \rangle)) \, x_i^T \Delta.$$

Applying the mean value theorem, we find that

$$\mathcal{E}_n(\Delta) = \frac{1}{n} \sum_{i=1}^n \psi''(\langle x_i, \, \beta^* \rangle + t_i \, \langle x_i, \, \Delta \rangle) \left(\langle x_i, \, \Delta \rangle \right)^2,$$

where $t_i \in [0, 1]$. From (the proof of) Proposition 2 in Negahban et al. (2012), we then have

$$\mathcal{E}_n(\Delta) \geq \alpha_1 \|\Delta\|_2^2 - \tau_1 \sqrt{\frac{\log p}{n}} \|\Delta\|_1 \|\Delta\|_2, \qquad \forall \|\Delta\|_2 \leq 1, \tag{62}$$

with probability at least $1 - c_1 \exp(-c_2 n)$, for an appropriate choice of α_1. Note that by the arithmetic mean-geometric mean inequality,

$$\tau_1 \sqrt{\frac{\log p}{n}} \|\Delta\|_1 \|\Delta\|_2 \leq \frac{\alpha_1}{2} \|\Delta\|_2^2 + \frac{\tau_1^2}{2\alpha_1} \frac{\log p}{n} \|\Delta\|_1^2,$$

and consequently,

$$\mathcal{E}_n(\Delta) \geq \frac{\alpha_1}{2}\|\Delta\|_2^2 - \frac{\tau_1^2}{2\alpha_1}\frac{\log p}{n}\|\Delta\|_1^2,$$

which establishes (4a). Inequality (4b) then follows via Lemma 8 in Appendix B.1.

It remains to show that there are universal constants (c, c_1, c_2) such that

$$\mathbb{P}\left(\|\nabla\mathcal{L}_n(\beta^*)\|_\infty \geq c\sqrt{\frac{\log p}{n}}\right) \leq c_1\exp(-c_2\log p). \tag{63}$$

For each $1 \leq i \leq n$ and $1 \leq j \leq p$, define the random variable $V_{ij} := (\psi'(x_i^T\beta^*) - y_i)x_{ij}$. Our goal is to bound $\max_{j=1,\ldots,p}|\frac{1}{n}\sum_{i=1}^n V_{ij}|$. Note that

$$\mathbb{P}\left[\max_{j=1,\ldots,p}\Big|\frac{1}{n}\sum_{i=1}^n V_{ij}\Big| \geq \delta\right] \leq \mathbb{P}[\mathcal{A}^c] + \mathbb{P}\left[\max_{j=1,\ldots,p}\Big|\frac{1}{n}\sum_{i=1}^n V_{ij}\Big| \geq \delta \;\Big|\; \mathcal{A}\right], \tag{64}$$

where

$$\mathcal{A} := \left\{\max_{j=1,\ldots,p}\left\{\frac{1}{n}\sum_{i=1}^n x_{ij}^2\right\} \leq 2\mathbb{E}[x_{ij}^2]\right\}.$$

Since the x_{ij}'s are sub-Gaussian and $n \gtrsim \log p$, there exist universal constants (c_1, c_2) such that $\mathbb{P}[\mathcal{A}^c] \leq c_1\exp(-c_2 n)$. The last step is to bound the second term on the right side of (64). For any $t \in \mathbb{R}$, we have

$$\begin{aligned}\log\mathbb{E}[\exp(tV_{ij}) \mid x_i] &= \log\left[\exp(tx_{ij}\psi'(x_i^T\beta^*)\right] \cdot \mathbb{E}[\exp(-tx_{ij}y_i)] \\ &= tx_{ij}\psi'(x_i^T\beta^*) + \left(\psi(-tx_{ij} + x_i^T\beta^*) - \psi(x_i^T\beta^*)\right),\end{aligned}$$

using the fact that ψ is the cumulant generating function for the underlying exponential family. Thus, by a Taylor series expansion, there is some $v_i \in [0,1]$ such that

$$\log\mathbb{E}[\exp(tV_{ij}) \mid x_i] = \frac{t^2x_{ij}^2}{2}\,\psi''(x_i^T\beta^* - v_i\,tx_{ij}) \;\leq\; \frac{\alpha_u t^2 x_{ij}^2}{2}, \tag{65}$$

where the inequality uses the boundedness of ψ''. Consequently, conditioned on the event $\mathcal{A}$, the variable $\frac{1}{n}\sum_{i=1}^n V_{ij}$ is sub-Gaussian with parameter at most $\kappa = \alpha_u \cdot \max_{j=1,\ldots,p}\mathbb{E}[x_{ij}^2]$, for each $j = 1, \ldots, p$. By a union bound, we then have

$$\mathbb{P}\left[\max_{j=1,\ldots,p}\Big|\frac{1}{n}\sum_{i=1}^n V_{ij}\Big| \geq \delta \;\Big|\; \mathcal{A}\right] \leq p\,\exp\left(-\frac{n\delta^2}{2\kappa^2}\right).$$

The claimed ℓ_1- and ℓ_2-bounds then follow directly from Theorem 1.

B.4 Proof of Corollary 3

We first verify condition (4a) in the case where $|||\Delta|||_F \leq 1$. A straightforward calculation yields

$$\nabla^2\mathcal{L}_n(\Theta) = \Theta^{-1} \otimes \Theta^{-1} = (\Theta \otimes \Theta)^{-1}.$$

Moreover, letting $\text{vec}(\Delta) \in \mathbb{R}^{p^2}$ denote the vectorized form of the matrix Δ, applying the mean value theorem yields

$$\mathcal{E}_n(\Delta) = \text{vec}(\Delta)^T \left(\nabla^2 \mathcal{L}_n(\Theta^* + t\Delta)\right) \text{vec}(\Delta) \geq \lambda_{\min}(\nabla^2 \mathcal{L}_n(\Theta^* + t\Delta)) \, |||\Theta|||_F^2, \tag{66}$$

for some $t \in [0, 1]$. By standard properties of the Kronecker product (Horn and Johnson, 1990), we have

$$\begin{aligned}\lambda_{\min}(\nabla^2 \mathcal{L}_n(\Theta^* + t\Delta)) = |||\Theta^* + t\Delta|||_2^{-2} &\geq (|||\Theta^*|||_2 + t \, |||\Delta|||_2)^{-2} \\ &\geq (|||\Theta^*|||_2 + 1)^{-2},\end{aligned}$$

using the fact that $|||\Delta|||_2 \leq |||\Delta|||_F \leq 1$. Plugging back into (66) yields

$$\mathcal{E}_n(\Delta) \geq (|||\Theta^*|||_2 + 1)^{-2} \, |||\Theta|||_F^2,$$

so (4a) holds with $\alpha_1 = (|||\Theta^*|||_2 + 1)^{-2}$ and $\tau_1 = 0$. Lemma 9 then implies (4b) with $\alpha_2 = (|||\Theta^*|||_2 + 1)^{-2}$. Finally, we need to establish that the given choice of λ satisfies the requirement (6) of Theorem 1. By the assumed deviation condition (17), we have

$$|||\nabla \mathcal{L}_n(\Theta^*)|||_{\max} = \left|\left|\left| \widehat{\Sigma} - (\Theta^*)^{-1} \right|\right|\right|_{\max} = \left|\left|\left| \widehat{\Sigma} - \Sigma \right|\right|\right|_{\max} \leq c_0 \sqrt{\frac{\log p}{n}}.$$

Applying Theorem 1 then implies the desired result.

Appendix C. Auxiliary Optimization-Theoretic Results

In this section, we provide proofs of the supporting lemmas used in Section 4.

C.1 Derivation of Three-Step Procedure

We begin by deriving the correctness of the three-step procedure given in Section 4.2. Let $\widehat{\beta}$ be the unconstrained optimum of the program (41). If $g_{\lambda,\mu}(\widehat{\beta}) \leq R$, we clearly have the update given in step (2). Suppose instead that $g_{\lambda,\mu}(\widehat{\beta}) > R$. Then since the program (31) is convex, the iterate β^{t+1} must lie on the boundary of the feasible set; i.e.,

$$g_{\lambda,\mu}(\beta^{t+1}) = R. \tag{67}$$

By Lagrangian duality, the program (31) is also equivalent to

$$\beta^{t+1} \in \arg \min_{g_{\lambda,\mu}(\beta) \leq R'} \left\{ \frac{1}{2} \left\| \beta - \left(\beta^t - \frac{\nabla \bar{\mathcal{L}}_n(\beta^t)}{\eta} \right) \right\|_2^2 \right\},$$

for some choice of constraint parameter R'. Note that this is projection of $\beta^t - \frac{\nabla \bar{\mathcal{L}}_n(\beta^t)}{\eta}$ onto the set $\{\beta \in \mathbb{R}^p \mid g_{\lambda,\mu}(\beta) \leq R'\}$. Since projection decreases the value of $g_{\lambda,\mu}$, equation (67) implies that

$$g_{\lambda,\mu}\left(\beta^t - \frac{\nabla \bar{\mathcal{L}}_n(\beta^t)}{\eta} \right) \geq R.$$

In fact, since the projection will shrink the vector to the boundary of the constraint set, (67) forces $R' = R$. This yields the update (42) appearing in step (3).

C.2 Derivation of Updates for SCAD and MCP

We now derive the explicit form of the updates (43) and (44) for the SCAD and MCP regularizers, respectively. We may rewrite the unconstrained program (41) as

$$\begin{aligned}\beta^{t+1} &\in \arg\min_{\beta\in\mathbb{R}^p}\left\{\frac{1}{2}\left\|\beta-\left(\beta^t-\frac{\nabla\bar{\mathcal{L}}_n(\beta^t)}{\eta}\right)\right\|_2^2+\frac{1}{\eta}\cdot\rho_\lambda(\beta)+\frac{\mu}{2\eta}\|\beta\|_2^2\right\}\\ &=\arg\min_{\beta\in\mathbb{R}^p}\left\{\left(\frac{1}{2}+\frac{\mu}{2\eta}\right)\|\beta\|_2^2-\beta^T\left(\beta^t-\frac{\nabla\bar{\mathcal{L}}_n(\beta^t)}{\eta}\right)+\frac{1}{\eta}\cdot\rho_\lambda(\beta)\right\}\\ &=\arg\min_{\beta\in\mathbb{R}^p}\left\{\frac{1}{2}\left\|\beta-\frac{1}{1+\mu/\eta}\left(\beta^t-\frac{\nabla\bar{\mathcal{L}}_n(\beta^t)}{\eta}\right)\right\|_2^2+\frac{1/\eta}{1+\mu/\eta}\cdot\rho_\lambda(\beta)\right\}. \qquad (68)\end{aligned}$$

Since the program in the last line of equation (68) decomposes by coordinate, it suffices to solve the scalar optimization problem

$$\widehat{x}\in\arg\min_x\left\{\frac{1}{2}(x-z)^2+\nu\rho(x;\lambda)\right\}, \qquad (69)$$

for general $z\in\mathbb{R}$ and $\nu>0$.

We first consider the case when ρ is the SCAD penalty. The solution $\widehat{x}$ of the program (69) in the case when $\nu=1$ is given in Fan and Li (2001); the expression (43) for the more general case comes from writing out the subgradient of the objective as

$$(x-z)+\nu\rho'(x;\lambda)=\begin{cases}(x-z)+\nu\lambda[-1,+1] & \text{if } x=0,\\ (x-z)+\nu\lambda & \text{if } 0<x\le\lambda,\\ (x-z)+\frac{\nu(a\lambda-x)}{a-1} & \text{if } \lambda\le x\le a\lambda,\\ x-z & \text{if } x\ge a\lambda,\end{cases}$$

using the equation for the SCAD derivative (58), and setting the subgradient equal to zero.

Similarly, when ρ is the MCP parameterized by (b,λ), the subgradient of the objective takes the form

$$(x-z)+\nu\rho'(x;\lambda)=\begin{cases}(x-z)+\nu\lambda[-1,+1] & \text{if } x=0,\\ (x-z)+\nu\lambda\left(1-\frac{x}{b\lambda}\right) & \text{if } 0<x\le b\lambda,\\ x-z & \text{if } x\ge b\lambda,\end{cases}$$

using the expression for the MCP derivative (59), leading to the closed-form solution given in (44). This agrees with the expression provided in Breheny and Huang (2011) for the special case when $\nu=1$.

C.3 Proof of Lemma 1

We first show that if $\lambda\ge\frac{8}{L}\cdot\|\nabla\mathcal{L}_n(\beta^*)\|_\infty$, then for any feasible β such that

$$\phi(\beta)\le\phi(\beta^*)+\bar{\eta}, \qquad (70)$$

we have

$$\|\beta - \beta^*\|_1 \leq 8\sqrt{k}\|\beta - \beta^*\|_2 + 2 \cdot \min\left(\frac{2\bar{\eta}}{\lambda L}, R\right). \tag{71}$$

Defining the error vector $\Delta := \beta - \beta^*$, (70) implies

$$\mathcal{L}_n(\beta^* + \Delta) + \rho_\lambda(\beta^* + \Delta) \leq \mathcal{L}_n(\beta^*) + \rho_\lambda(\beta^*) + \bar{\eta},$$

so subtracting $\langle \nabla \mathcal{L}_n(\beta^*), \Delta \rangle$ from both sides gives

$$\mathcal{T}(\beta^* + \Delta, \beta^*) + \rho_\lambda(\beta^* + \Delta) - \rho_\lambda(\beta^*) \leq -\langle \nabla \mathcal{L}_n(\beta^*), \Delta \rangle + \bar{\eta}. \tag{72}$$

We divide the argument into two cases. First suppose $\|\Delta\|_2 \leq 3$. Note that if $\bar{\eta} \geq \frac{\lambda L}{4}\|\Delta\|_1$, the claim (71) is trivially true; so assume $\bar{\eta} \leq \frac{\lambda L}{4}\|\Delta\|_1$. Then the RSC condition (34a), together with (72), implies that

$$\begin{aligned}\alpha_1\|\Delta\|_2^2 - \tau_1\frac{\log p}{n}\|\Delta\|_1^2 + \rho_\lambda(\beta^* + \Delta) - \rho_\lambda(\beta^*) &\leq \|\nabla \mathcal{L}_n(\beta^*)\|_\infty \cdot \|\Delta\|_1 + \bar{\eta} \\ &\leq \frac{\lambda L}{8}\|\Delta\|_1 + \frac{\lambda L}{4}\|\Delta\|_1. \end{aligned} \tag{73}$$

Rearranging and using the assumption $\lambda L \geq 16R\tau_1\frac{\log p}{n}$, along with Lemma 4 in Appendix A.1, we then have

$$\begin{aligned}\alpha_1\|\Delta\|_2^2 &\leq \rho_\lambda(\beta^*) - \rho_\lambda(\beta^* + \Delta) + \frac{\lambda L}{2}\|\Delta\|_1 \\ &\leq \rho_\lambda(\beta^*) - \rho_\lambda(\beta^* + \Delta) + \frac{\rho_\lambda(\beta^*) + \rho_\lambda(\beta^* + \Delta)}{2} + \frac{\mu}{4}\|\Delta\|_2^2,\end{aligned}$$

implying that

$$0 \leq \left(\alpha_1 - \frac{\mu}{4}\right)\|\Delta\|_2^2 \leq \frac{3}{2}\rho_\lambda(\beta^*) - \frac{1}{2}\rho_\lambda(\beta^* + \Delta),$$

so

$$\rho_\lambda(\beta^*) - \rho_\lambda(\beta^* + \Delta) \leq 3\rho_\lambda(\beta^*) - \rho_\lambda(\beta^* + \Delta) \leq 3\lambda L\|\Delta_A\|_1 - \lambda L\|\Delta_{A^c}\|_1, \tag{74}$$

by Lemma 5 in Appendix A.1. Furthermore, note that the bound (73) also implies that

$$\rho_\lambda(\beta^* + \Delta) - \rho_\lambda(\beta^*) \leq \frac{\lambda L}{2}\|\Delta\|_1 + \bar{\eta}. \tag{75}$$

Combining (74) and (75) then gives

$$\|\Delta_{A^c}\|_1 - 3\|\Delta_A\|_1 \leq \frac{1}{2}\|\Delta\|_1 + \frac{\bar{\eta}}{\lambda L} \leq \frac{1}{2}\|\Delta_A\|_1 + \frac{1}{2}\|\Delta_{A^c}\|_1 + \frac{\bar{\eta}}{\lambda L},$$

so

$$\|\Delta_{A^c}\|_1 \leq 7\|\Delta_A\|_1 + \frac{2\bar{\eta}}{\lambda L},$$

implying that

$$\|\Delta\|_1 \leq 8\|\Delta_A\|_1 + \frac{2\bar{\eta}}{\lambda L} \leq 8\sqrt{k}\|\Delta\|_2 + \frac{2\bar{\eta}}{\lambda L}.$$

In the case when $\|\Delta\|_2 \geq 3$, the RSC condition (34b) gives

$$\alpha_2\|\Delta\|_2 - \tau_2\sqrt{\frac{\log p}{n}}\|\Delta\|_1 + \rho_\lambda(\beta^* + \Delta) - \rho_\lambda(\beta^*) \leq \|\nabla\mathcal{L}_n(\beta^*)\|_\infty \cdot \|\Delta\|_1 + \bar{\eta}$$
$$\leq \frac{\lambda L}{8}\|\Delta\|_1 + \frac{\lambda L}{4}\|\Delta\|_1, \tag{76}$$

so

$$\alpha_2\|\Delta\|_2 \leq \rho_\lambda(\beta^*) - \rho_\lambda(\beta^* + \Delta) + \left(\frac{3\lambda L}{8} + \tau_2\sqrt{\frac{\log p}{n}}\right)\|\Delta\|_1.$$

In particular, if $\rho_\lambda(\beta^*) - \rho_\lambda(\beta^* + \Delta) \leq 0$, we have

$$\|\Delta\|_2 \leq \frac{2R}{\alpha_2}\left(\frac{3\lambda L}{8} + \tau_2\sqrt{\frac{\log p}{n}}\right) < 3,$$

a contradiction. Hence, using Lemma 5 in Appendix A.1, we have

$$0 \leq \rho_\lambda(\beta^*) - \rho_\lambda(\beta^* + \Delta) \leq \lambda L\|\Delta_A\|_1 - \lambda L\|\Delta_{A^c}\|_1. \tag{77}$$

Note that under the scaling $\lambda L \geq 4\tau_2\sqrt{\frac{\log p}{n}}$, the bound (76) also implies (75). Combining (75) and (77), we then have

$$\|\Delta_{A^c}\|_1 - \|\Delta_A\|_1 \leq \frac{1}{2}\|\Delta\|_1 + \frac{\bar{\eta}}{\lambda L} = \frac{1}{2}\|\Delta_{A^c}\|_1 + \frac{1}{2}\|\Delta_A\|_1 + \frac{\bar{\eta}}{\lambda L},$$

and consequently,

$$\|\Delta_{A^c}\|_1 \leq 3\|\Delta_A\|_1 + \frac{2\bar{\eta}}{\lambda L},$$

so

$$\|\Delta\|_1 \leq 4\|\Delta_A\|_1 + \frac{2\bar{\eta}}{\lambda L} \leq 4\sqrt{k}\|\Delta\|_2 + \frac{2\bar{\eta}}{\lambda L}.$$

Using the trivial bound $\|\Delta\|_1 \leq 2R$, we obtain the claim (71).

We now apply the implication (70) to the vectors $\widehat{\beta}$ and β^t. Note that by optimality of $\widehat{\beta}$, we have

$$\phi(\widehat{\beta}) \leq \phi(\beta^*),$$

and by the assumption (45), we also have

$$\phi(\beta^t) \leq \phi(\widehat{\beta}) + \bar{\eta} \leq \phi(\beta^*) + \bar{\eta}.$$

Hence,

$$\|\widehat{\beta} - \beta^*\|_1 \leq 8\sqrt{k}\|\widehat{\beta} - \beta^*\|_2, \qquad \text{and}$$
$$\|\beta^t - \beta^*\|_1 \leq 8\sqrt{k}\|\beta^t - \beta^*\|_2 + 2 \cdot \min\left(\frac{2\bar{\eta}}{\lambda L}, R\right).$$

By the triangle inequality, we then have

$$\begin{aligned}\|\beta^t - \widehat{\beta}\|_1 &\leq \|\widehat{\beta} - \beta^*\|_1 + \|\beta^t - \beta^*\|_1 \\ &\leq 8\sqrt{k} \cdot \Big(\|\widehat{\beta} - \beta^*\|_2 + \|\beta^t - \beta^*\|_2\Big) + 2 \cdot \min\left(\frac{2\overline{\eta}}{\lambda L}, R\right) \\ &\leq 8\sqrt{k} \cdot \Big(2\|\widehat{\beta} - \beta^*\|_2 + \|\beta^t - \widehat{\beta}\|_2\Big) + 2 \cdot \min\left(\frac{2\overline{\eta}}{\lambda L}, R\right),\end{aligned}$$

as claimed.

C.4 Proof of Lemma 2

Our proof proceeds via induction on the iteration number t. Note that the base case $t = 0$ holds by assumption. Hence, it remains to show that if $\|\beta^t - \widehat{\beta}\|_2 \leq 3$ for some integer $t \geq 1$, then $\|\beta^{t+1} - \widehat{\beta}\|_2 \leq 3$, as well.

We assume for the sake of a contradiction that $\|\beta^{t+1} - \widehat{\beta}\|_2 > 3$. By the RSC condition (34b) and the relation (33), we have

$$\overline{\mathcal{T}}(\beta^{t+1}, \widehat{\beta}) \geq \alpha\|\widehat{\beta} - \beta^{t+1}\|_2 - \tau\sqrt{\frac{\log p}{n}}\|\widehat{\beta} - \beta^{t+1}\|_1 - \frac{\mu}{2}\|\widehat{\beta} - \beta^{t+1}\|_2^2. \tag{78}$$

Furthermore, by convexity of $g := g_{\lambda,\mu}$, we have

$$g(\beta^{t+1}) - g(\widehat{\beta}) - \langle \nabla g(\widehat{\beta}),\, \beta^{t+1} - \widehat{\beta}\rangle \geq 0. \tag{79}$$

Multiplying by λ and summing with (78) then yields

$$\begin{aligned}&\phi(\beta^{t+1}) - \phi(\widehat{\beta}) - \langle \nabla\phi(\widehat{\beta}),\, \beta^{t+1} - \widehat{\beta}\rangle \\ &\qquad \geq \alpha\|\widehat{\beta} - \beta^{t+1}\|_2 - \tau\sqrt{\frac{\log p}{n}}\|\widehat{\beta} - \beta^{t+1}\|_1 - \frac{\mu}{2}\|\widehat{\beta} - \beta^{t+1}\|_2^2.\end{aligned}$$

Together with the first-order optimality condition $\langle \nabla\phi(\widehat{\beta}),\, \beta^{t+1} - \widehat{\beta}\rangle \geq 0$, we then have

$$\phi(\beta^{t+1}) - \phi(\widehat{\beta}) \geq \alpha\|\widehat{\beta} - \beta^{t+1}\|_2 - \tau\sqrt{\frac{\log p}{n}}\|\widehat{\beta} - \beta^{t+1}\|_1 - \frac{\mu}{2}\|\widehat{\beta} - \beta^{t+1}\|_2^2. \tag{80}$$

Since $\|\widehat{\beta} - \beta^t\|_2 \leq 3$ by the induction hypothesis, applying the RSC condition (34a) to the pair $(\widehat{\beta}, \beta^t)$ also gives

$$\overline{\mathcal{L}}_n(\widehat{\beta}) \geq \overline{\mathcal{L}}_n(\beta^t) + \langle \nabla\overline{\mathcal{L}}_n(\beta^t),\, \widehat{\beta} - \beta^t\rangle + \Big(\alpha - \frac{\mu}{2}\Big) \cdot \|\beta^t - \widehat{\beta}\|_2^2 - \tau\frac{\log p}{n}\|\beta^t - \widehat{\beta}\|_1^2.$$

Combining with the inequality

$$g(\widehat{\beta}) \geq g(\beta^{t+1}) + \langle \nabla g(\beta^{t+1}),\, \widehat{\beta} - \beta^{t+1}\rangle,$$

we then have

$$\begin{aligned}\phi(\widehat{\beta}) &\geq \bar{\mathcal{L}}_n(\beta^t) + \langle \nabla \bar{\mathcal{L}}_n(\beta^t),\, \widehat{\beta} - \beta^t \rangle + \lambda g(\beta^{t+1}) + \lambda \langle \nabla g(\beta^{t+1}),\, \widehat{\beta} - \beta^{t+1} \rangle \\ &\qquad + \left(\alpha - \frac{\mu}{2}\right) \cdot \|\beta^t - \widehat{\beta}\|_2^2 - \tau \frac{\log p}{n} \|\beta^t - \widehat{\beta}\|_1^2 \\ &\geq \bar{\mathcal{L}}_n(\beta^t) + \langle \nabla \bar{\mathcal{L}}_n(\beta^t),\, \widehat{\beta} - \beta^t \rangle + \lambda g(\beta^{t+1}) \\ &\qquad + \lambda \langle \nabla g(\beta^{t+1}),\, \widehat{\beta} - \beta^{t+1} \rangle - \tau \frac{\log p}{n} \|\beta^t - \widehat{\beta}\|_1^2. \end{aligned} \tag{81}$$

Finally, the RSM condition (35) on the pair (β^{t+1}, β^t) gives

$$\begin{aligned}\phi(\beta^{t+1}) &\leq \bar{\mathcal{L}}_n(\beta^t) + \langle \nabla \bar{\mathcal{L}}_n(\beta^t),\, \beta^{t+1} - \beta^t \rangle + \lambda g(\beta^{t+1}) \qquad (82) \\ &\qquad + \left(\alpha_3 - \frac{\mu}{2}\right) \|\beta^{t+1} - \beta^t\|_2^2 + \tau \frac{\log p}{n} \|\beta^{t+1} - \beta^t\|_1^2 \\ &\leq \bar{\mathcal{L}}_n(\beta^t) + \langle \nabla \bar{\mathcal{L}}_n(\beta^t),\, \beta^{t+1} - \beta^t \rangle + \lambda g(\beta^{t+1}) \\ &\qquad + \frac{\eta}{2} \|\beta^{t+1} - \beta^t\|_2^2 + \frac{4R^2 \tau \log p}{n}, \qquad (83) \end{aligned}$$

since $\frac{\eta}{2} \geq \alpha_3 - \frac{\mu}{2}$ by assumption, and $\|\beta^{t+1} - \beta^t\|_1 \leq 2R$. It is easy to check that the update (31) may be written equivalently as

$$\beta^{t+1} \in \arg \min_{g(\beta) \leq R,\, \beta \in \Omega} \left\{ \bar{\mathcal{L}}_n(\beta^t) + \langle \nabla \bar{\mathcal{L}}_n(\beta^t),\, \beta - \beta^t \rangle + \frac{\eta}{2} \|\beta - \beta^t\|_2^2 + \lambda g(\beta) \right\},$$

and the optimality of β^{t+1} then yields

$$\langle \nabla \bar{\mathcal{L}}_n(\beta^t) + \eta(\beta^{t+1} - \beta^t) + \lambda \nabla g(\beta^{t+1}),\, \beta^{t+1} - \widehat{\beta} \rangle \leq 0. \tag{84}$$

Summing up (81), (82), and (84), we then have

$$\begin{aligned}\phi(\beta^{t+1}) - \phi(\widehat{\beta}) &\leq \frac{\eta}{2} \|\beta^{t+1} - \beta^t\|_2^2 + \eta \langle \beta^t - \beta^{t+1},\, \beta^{t+1} - \widehat{\beta} \rangle + \tau \frac{\log p}{n} \|\beta^t - \widehat{\beta}\|_1^2 \\ &\qquad + \frac{4R^2 \tau \log p}{n} \\ &= \frac{\eta}{2} \|\beta^t - \widehat{\beta}\|_2^2 - \frac{\eta}{2} \|\beta^{t+1} - \widehat{\beta}\|_2^2 + \tau \frac{\log p}{n} \|\beta^t - \widehat{\beta}\|_1^2 + \frac{4R^2 \tau \log p}{n}. \end{aligned}$$

Combining this last inequality with (80), we have

$$\begin{aligned}\alpha \|\widehat{\beta} - \beta^{t+1}\|_2 - \tau \sqrt{\frac{\log p}{n}} \|\widehat{\beta} - \beta^{t+1}\|_1 & \\ \leq \frac{\eta}{2} \|\beta^t - \widehat{\beta}\|_2^2 - \frac{\eta - \mu}{2} \|\beta^{t+1} - \widehat{\beta}\|_2^2 + \frac{8R^2 \tau \log p}{n} & \\ \leq \frac{9\eta}{2} - \frac{3(\eta - \mu)}{2} \|\beta^{t+1} - \widehat{\beta}\|_2 + \frac{8R^2 \tau \log p}{n}, & \end{aligned}$$

since $\|\beta^t - \widehat{\beta}\|_2 \leq 3$ by the induction hypothesis and $\|\beta^{t+1} - \widehat{\beta}\|_2 > 3$ by assumption, and using the fact that $\eta \geq \mu$. It follows that

$$\begin{aligned}\left(\alpha + \frac{3(\eta-\mu)}{2}\right) \cdot \|\widehat{\beta} - \beta^{t+1}\|_2 &\leq \frac{9\eta}{2} + \tau\sqrt{\frac{\log p}{n}}\|\widehat{\beta} - \beta^{t+1}\|_1 + \frac{8R^2\tau\log p}{n} \\ &\leq \frac{9\eta}{2} + 2R\tau\sqrt{\frac{\log p}{n}} + \frac{8R^2\tau\log p}{n} \\ &\leq 3\left(\alpha + \frac{3(\eta-\mu)}{2}\right),\end{aligned}$$

where the final inequality holds whenever $2R\tau\sqrt{\frac{\log p}{n}} + \frac{8R^2\tau\log p}{n} \leq 3\left(\alpha - \frac{3\mu}{2}\right)$. Rearranging gives $\|\beta^{t+1} - \widehat{\beta}\|_2 \leq 3$, providing the desired contradiction.

C.5 Proof of Lemma 3

We begin with an auxiliary lemma:

Lemma 10 *Under the conditions of Lemma 3, we have*

$$\overline{\mathcal{T}}(\beta^t, \widehat{\beta}) \geq -2\tau\frac{\log p}{n}(\epsilon + \bar{\epsilon})^2, \quad \text{and} \tag{85a}$$

$$\phi(\beta^t) - \phi(\widehat{\beta}) \geq \frac{2\alpha - \mu}{4}\|\widehat{\beta} - \beta^t\|_2^2 - \frac{2\tau\log p}{n}(\epsilon + \bar{\epsilon})^2. \tag{85b}$$

We prove this result later, taking it as given for the moment.

Define

$$\phi_t(\beta) := \bar{\mathcal{L}}_n(\beta^t) + \langle \nabla\bar{\mathcal{L}}_n(\beta^t),\, \beta - \beta^t\rangle + \frac{\eta}{2}\|\beta - \beta^t\|_2^2 + \lambda g(\beta),$$

the objective function minimized over the constraint set $\{g(\beta) \leq R\}$ at iteration t. For any $\gamma \in [0,1]$, the vector $\beta_\gamma := \gamma\widehat{\beta} + (1-\gamma)\beta^t$ belongs to the constraint set, as well. Consequently, by the optimality of β^{t+1} and feasibility of β_γ, we have

$$\phi_t(\beta^{t+1}) \leq \phi_t(\beta_\gamma) = \bar{\mathcal{L}}_n(\beta^t) + \langle \nabla\bar{\mathcal{L}}_n(\beta^t),\, \gamma\widehat{\beta} - \gamma\beta^t\rangle + \frac{\eta\gamma^2}{2}\|\widehat{\beta} - \beta^t\|_2^2 + \lambda g(\beta_\gamma).$$

Appealing to (85a), we then have

$$\begin{aligned}\phi_t(\beta^{t+1}) &\leq (1-\gamma)\bar{\mathcal{L}}_n(\beta^t) + \gamma\bar{\mathcal{L}}_n(\widehat{\beta}) + 2\gamma\tau\frac{\log p}{n}(\epsilon + \bar{\epsilon})^2 \\ &\qquad + \frac{\eta\gamma^2}{2}\|\widehat{\beta} - \beta^t\|_2^2 + \lambda g(\beta_\gamma) \\ &\overset{(i)}{\leq} \phi(\beta^t) - \gamma(\phi(\beta^t) - \phi(\widehat{\beta})) + 2\gamma\tau\frac{\log p}{n}(\epsilon + \bar{\epsilon})^2 + \frac{\eta\gamma^2}{2}\|\widehat{\beta} - \beta^t\|_2^2 \\ &\leq \phi(\beta^t) - \gamma(\phi(\beta^t) - \phi(\widehat{\beta})) + 2\tau\frac{\log p}{n}(\epsilon + \bar{\epsilon})^2 + \frac{\eta\gamma^2}{2}\|\widehat{\beta} - \beta^t\|_2^2,\end{aligned} \tag{86}$$

where inequality (i) incorporates the fact that

$$g(\beta_\gamma) \leq \gamma g(\widehat{\beta}) + (1-\gamma) g(\beta^t),$$

by the convexity of g.

By the RSM condition (35), we also have

$$\overline{\mathcal{T}}(\beta^{t+1}, \beta^t) \leq \frac{\eta}{2}\|\beta^{t+1} - \beta^t\|_2^2 + \tau\frac{\log p}{n}\|\beta^{t+1} - \beta^t\|_1^2,$$

since $\alpha_3 - \mu \leq \frac{\eta}{2}$ by assumption, and adding $\lambda g(\beta^{t+1})$ to both sides gives

$$\begin{aligned}\phi(\beta^{t+1}) &\leq \bar{\mathcal{L}}_n(\beta^t) + \langle \nabla \bar{\mathcal{L}}_n(\beta^t),\, \beta^{t+1} - \beta^t\rangle + \frac{\eta}{2}\|\beta^{t+1} - \beta^t\|_2^2 \\ &\qquad + \tau\frac{\log p}{n}\|\beta^{t+1} - \beta^t\|_1^2 + \lambda g(\beta^{t+1}) \\ &= \phi_t(\beta^{t+1}) + \tau\frac{\log p}{n}\|\beta^{t+1} - \beta^t\|_1^2.\end{aligned}$$

Combining with (86) then yields

$$\begin{aligned}\phi(\beta^{t+1}) \leq \phi(\beta^t) - \gamma(\phi(\beta^t) - \phi(\widehat{\beta})) + \frac{\eta\gamma^2}{2}\|\widehat{\beta} - \beta^t\|_2^2 \\ + \tau\frac{\log p}{n}\|\beta^{t+1} - \beta^t\|_1^2 + 2\tau\frac{\log p}{n}(\epsilon + \bar{\epsilon})^2. \qquad (87)\end{aligned}$$

By the triangle inequality, we have

$$\|\beta^{t+1} - \beta^t\|_1^2 \leq \left(\|\Delta^{t+1}\|_1 + \|\Delta^t\|_1\right)^2 \;\leq\; 2\|\Delta^{t+1}\|_1^2 + 2\|\Delta^t\|_1^2,$$

where we have defined $\Delta^t := \beta^t - \widehat{\beta}$. Combined with (87), we therefore have

$$\begin{aligned}\phi(\beta^{t+1}) \leq \phi(\beta^t) - \gamma(\phi(\beta^t) - \phi(\widehat{\beta})) + \frac{\eta\gamma^2}{2}\|\Delta^t\|_2^2 \\ + 2\tau\frac{\log p}{n}(\|\Delta^{t+1}\|_1^2 + \|\Delta^t\|_1^2) + 2\psi(n,p,\epsilon),\end{aligned}$$

where $\psi(n,p,\epsilon) := \tau\frac{\log p}{n}(\epsilon + \bar{\epsilon})^2$. Then applying Lemma 1 to bound the ℓ_1-norms, we have

$$\begin{aligned}\phi(\beta^{t+1}) &\leq \phi(\beta^t) - \gamma(\phi(\beta^t) - \phi(\widehat{\beta})) + \frac{\eta\gamma^2}{2}\|\Delta^t\|_2^2 \\ &\qquad + ck\tau\frac{\log p}{n}(\|\Delta^{t+1}\|_2^2 + \|\Delta^t\|_2^2) + c'\psi(n,p,\epsilon) \\ &= \phi(\beta^t) - \gamma(\phi(\beta^t) - \phi(\widehat{\beta})) + \left(\frac{\eta\gamma^2}{2} + ck\tau\frac{\log p}{n}\right)\|\Delta^t\|_2^2 \\ &\qquad + ck\tau\frac{\log p}{n}\|\Delta^{t+1}\|_2^2 + c'\psi(n,p,\epsilon). \qquad (88)\end{aligned}$$

Now introduce the shorthand $\delta_t := \phi(\beta^t) - \phi(\widehat{\beta})$ and $\upsilon(k,p,n) = k\tau\frac{\log p}{n}$. By applying (85b) and subtracting $\phi(\widehat{\beta})$ from both sides of (88), we have

$$\delta_{t+1} \leq \big(1-\gamma\big)\delta_t + \frac{\eta\gamma^2 + c\upsilon(k,p,n)}{\alpha - \mu/2}\left(\delta_t + 2\psi(n,p,\epsilon)\right) + \frac{c\upsilon(k,p,n)}{\alpha-\mu/2}\left(\delta_{t+1} + 2\psi(n,p,\epsilon)\right) + c'\psi(n,p,\epsilon).$$

Choosing $\gamma = \frac{2\alpha-\mu}{4\eta} \in (0,1)$ yields

$$\left(1 - \frac{c\upsilon(k,p,n)}{\alpha-\mu/2}\right)\delta_{t+1} \leq \left(1 - \frac{2\alpha-\mu}{8\eta} + \frac{c\upsilon(k,p,n)}{\alpha-\mu/2}\right)\delta_t + 2\left(\frac{2\alpha-\mu}{8\eta} + \frac{2c\upsilon(k,p,n)}{\alpha-\mu/2} + c'\right)\psi(n,p,\epsilon),$$

or $\delta_{t+1} \leq \kappa\delta_t + \xi(\epsilon+\bar{\epsilon})^2$, where κ and ξ were previously defined in (36) and (47), respectively. Finally, iterating the procedure yields

$$\delta_t \leq \kappa^{t-T}\delta_T + \xi(\epsilon+\bar{\epsilon})^2(1+\kappa+\kappa^2+\cdots+\kappa^{t-T-1}) \;\leq\; \kappa^{t-T}\delta_T + \frac{\xi(\epsilon+\bar{\epsilon})^2}{1-\kappa}, \tag{89}$$

as claimed.

The only remaining step is to prove the auxiliary lemma.

Proof of Lemma 10: By the RSC condition (34a) and the assumption (46), we have

$$\overline{\mathcal{T}}(\beta^t, \widehat{\beta}) \geq \left(\alpha - \frac{\mu}{2}\right)\|\widehat{\beta} - \beta^t\|_2^2 - \tau\frac{\log p}{n}\|\widehat{\beta} - \beta^t\|_1^2. \tag{90}$$

Furthermore, by convexity of g, we have

$$\lambda\Big(g(\beta^t) - g(\widehat{\beta}) - \langle\nabla g(\widehat{\beta}),\, \beta^t - \widehat{\beta}\rangle\Big) \geq 0, \tag{91}$$

and the first-order optimality condition for $\widehat{\beta}$ gives

$$\langle\nabla\phi(\widehat{\beta}),\, \beta^t - \widehat{\beta}\rangle \geq 0. \tag{92}$$

Summing (90), (91), and (92) then yields

$$\phi(\beta^t) - \phi(\widehat{\beta}) \geq \left(\alpha - \frac{\mu}{2}\right)\|\widehat{\beta} - \beta^t\|_2^2 - \tau\frac{\log p}{n}\|\widehat{\beta} - \beta^t\|_1^2.$$

Applying Lemma 1 to bound the term $\|\widehat{\beta} - \beta^t\|_1^2$ and using the assumption $\frac{ck\tau\log p}{n} \leq \frac{2\alpha-\mu}{4}$ yields the bound (85b). On the other hand, applying Lemma 1 directly to (90) with β^t and $\widehat{\beta}$ switched gives

$$\begin{aligned}\overline{\mathcal{T}}(\widehat{\beta}, \beta^t) &\geq \left(\alpha - \frac{\mu}{2}\right)\|\widehat{\beta} - \beta^t\|_2^2 - \tau\frac{\log p}{n}\left(ck\|\widehat{\beta} - \beta^t\|_2^2 + 2(\epsilon+\bar{\epsilon})^2\right) \\ &\geq -2\tau\frac{\log p}{n}(\epsilon+\bar{\epsilon})^2.\end{aligned}$$

This establishes (85a).

Appendix D. Verifying RSC/RSM Conditions

In this Appendix, we provide a proof of Proposition 1, which verifies the RSC (34) and RSM (35) conditions for GLMs.

D.1 Main Argument

Using the notation for GLMs in Section 3.3, we introduce the shorthand $\Delta := \beta_1 - \beta_2$ and observe that, by the mean value theorem, we have

$$\mathcal{T}(\beta_1, \beta_2) = \frac{1}{n} \sum_{i=1}^{n} \psi''\big(\langle \beta_1, x_i \rangle) + t_i \langle \Delta, x_i \rangle\big) \, (\langle \Delta, x_i \rangle)^2, \tag{93}$$

for some $t_i \in [0, 1]$. The t_i's are i.i.d. random variables, with each t_i depending only on the random vector x_i.

Proof of bound (40)**:** The proof of this upper bound is relatively straightforward given earlier results (Loh and Wainwright, 2013a). From the Taylor series expansion (93) and the boundedness assumption $\|\psi''\|_\infty \leq \alpha_u$, we have

$$\mathcal{T}(\beta_1, \beta_2) \leq \alpha_u \cdot \frac{1}{n} \sum_{i=1}^{n} \big(\langle \Delta, x_i \rangle\big)^2.$$

By known results on restricted eigenvalues for ordinary linear regression (cf. Lemma 13 in Loh and Wainwright (2012)), we also have

$$\frac{1}{n} \sum_{i=1}^{n} (\langle \Delta, x_i \rangle)^2 \leq \lambda_{\max}(\Sigma) \left(\frac{3}{2} \|\Delta\|_2^2 + \frac{\log p}{n} \|\Delta\|_1^2 \right),$$

with probability at least $1 - c_1 \exp(-c_2 n)$. Combining the two inequalities yields the desired result.

Proof of bounds (39)**:** The proof of the RSC bound is much more involved, and we provide only high-level details here, deferring the bulk of the technical analysis to later in the appendix. We define

$$\alpha_\ell := \left(\inf_{|t| \leq 2T} \psi''(t) \right) \frac{\lambda_{\min}(\Sigma)}{8},$$

where T is a suitably chosen constant depending only on $\lambda_{\min}(\Sigma)$ and the sub-Gaussian parameter σ_x. (In particular, see (99) below, and take $T = 3\tau$.) The core of the proof is based on the following lemma, proved in Section D.2:

Lemma 11 *With probability at least* $1 - c_1 \exp(-c_2 n)$*, we have*

$$\mathcal{T}(\beta_1, \beta_2) \geq \alpha_\ell \|\Delta\|_2^2 - c \sigma_x \|\Delta\|_1 \|\Delta\|_2 \sqrt{\frac{\log p}{n}},$$

uniformly over all pairs (β_1, β_2) *such that* $\beta_2 \in \mathbb{B}_2(3) \cap \mathbb{B}_1(R)$, $\|\beta_1 - \beta_2\|_2 \leq 3$, *and*

$$\frac{\|\Delta\|_1}{\|\Delta\|_2} \leq \frac{\alpha_\ell}{c\sigma_x}\sqrt{\frac{n}{\log p}}. \tag{94}$$

Taking Lemma 11 as given, we now complete the proof of the RSC condition (39). By the arithmetic mean-geometric mean inequality, we have

$$c\sigma_x\|\Delta\|_1\|\Delta\|_2\sqrt{\frac{\log p}{n}} \leq \frac{\alpha_\ell}{2}\|\Delta\|_2^2 + \frac{c^2\sigma_x^2}{2\alpha_\ell}\frac{\log p}{n}\|\Delta\|_1^2,$$

so Lemma 11 implies that (39a) holds uniformly over all pairs (β_1, β_2) such that $\beta_2 \in \mathbb{B}_2(3) \cap \mathbb{B}_1(R)$ and $\|\beta_1 - \beta_2\|_2 \leq 3$, whenever the bound (94) holds. On the other hand, if the bound (94) does not hold, then the lower bound in (39a) is negative. By convexity of $\mathcal{L}_n$, we have $\mathcal{T}(\beta_1, \beta_2) \geq 0$, so (39a) holds trivially in that case.

We now show that (39b) holds: in particular, consider a pair (β_1, β_2) with $\beta_2 \in \mathbb{B}_2(3)$ and $\|\beta_1 - \beta_2\|_2 \geq 3$. For any $t \in [0, 1]$, the convexity of $\mathcal{L}_n$ implies that

$$\mathcal{L}_n(\beta_2 + t\Delta) \leq t\mathcal{L}_n(\beta_2 + \Delta) + (1 - t)\mathcal{L}_n(\beta_2),$$

where $\Delta := \beta_1 - \beta_2$. Rearranging yields

$$\mathcal{L}_n(\beta_2 + \Delta) - \mathcal{L}_n(\beta_2) \geq \frac{\mathcal{L}_n(\beta_2 + t\Delta) - \mathcal{L}_n(\beta_2)}{t},$$

so

$$\mathcal{T}(\beta_2 + \Delta, \beta_2) \geq \frac{\mathcal{T}(\beta_2 + t\Delta, \beta_2)}{t}. \tag{95}$$

Now choose $t = \frac{3}{\|\Delta\|_2} \in [0, 1]$ so that $\|t\Delta\|_2 = 1$. Introducing the shorthand $\alpha_1 := \frac{\alpha_\ell}{2}$ and $\tau_1 := \frac{c^2\sigma_x^2}{2\alpha_\ell}$, we may apply (39a) to obtain

$$\begin{aligned}\frac{\mathcal{T}(\beta_2 + t\Delta, \beta_2)}{t} &\geq \frac{\|\Delta\|_2}{3}\left(\alpha_1\left(\frac{3\|\Delta\|_2}{\|\Delta\|_2}\right)^2 - \tau_1\frac{\log p}{n}\left(\frac{3\|\Delta\|_1}{\|\Delta\|_2}\right)^2\right) \\ &= 3\alpha_1\|\Delta\|_2 - 9\tau_1\frac{\log p}{n}\frac{\|\Delta\|_1^2}{\|\Delta\|_2}. \end{aligned} \tag{96}$$

Note that (39b) holds trivially unless $\frac{\|\Delta\|_1}{\|\Delta\|_2} \leq \frac{\alpha_\ell}{2c\sigma_x}\sqrt{\frac{n}{\log p}}$, due to the convexity of $\mathcal{L}_n$. In that case, (95) and (96) together imply

$$\mathcal{T}(\beta_2 + \Delta, \beta_2) \geq 3\alpha_1\|\Delta\|_2 - \frac{9\tau_1\,\alpha_\ell}{2c\sigma_x}\sqrt{\frac{\log p}{n}}\|\Delta\|_1,$$

which is exactly the bound (39b).

D.2 Proof of Lemma 11

For a truncation level $\tau' > 0$ to be chosen, define the functions

$$\varphi_{\tau'}(u) = \begin{cases} u^2, & \text{if } |u| \leq \frac{\tau'}{2}, \\ (\tau' - u)^2, & \text{if } \frac{\tau'}{2} \leq |u| \leq \tau', \\ 0, & \text{if } |u| \geq \tau'. \end{cases}$$

By construction, $\varphi_{\tau'}$ is τ'-Lipschitz and

$$\varphi_{\tau'}(u) \leq u^2 \cdot \mathbb{I}\{|u| \leq \tau'\}, \quad \text{for all } u \in \mathbb{R}. \tag{97}$$

In addition, we define the trapezoidal function

$$\gamma'_{\tau}(u) = \begin{cases} 1, & \text{if } |u| \leq \frac{\tau'}{2}, \\ 2 - \frac{2}{\tau'}|u|, & \text{if } \frac{\tau'}{2} \leq |u| \leq \tau', \\ 0, & \text{if } |u| \geq \tau', \end{cases}$$

and note that γ'_τ is $\frac{2}{\tau'}$-Lipschitz and $\gamma'_\tau(u) \leq \mathbb{I}\{|u| \leq \tau'\}$.

Taking $T \geq 3\tau'$ so that $T \geq \tau'\|\Delta\|_2$ (since $\|\Delta\|_2 \leq 3$ by assumption), and defining

$$L_\psi(T) := \inf_{|u| \leq 2T} \psi''(u),$$

we have the following inequality:

$$\begin{aligned} \mathcal{T}(\beta + \Delta, \beta) &= \frac{1}{n} \sum_{i=1}^n \psi''(x_i^T\beta + t_i \cdot x_i^T\Delta) \cdot (x_i^T\Delta)^2 \\ &\geq L_\psi(T) \cdot \sum_{i=1}^n (x_i^T\Delta)^2 \cdot \mathbb{I}\{|x_i^T\Delta| \leq \tau'\|\Delta\|_2\} \cdot \mathbb{I}\{|x_i^T\beta| \leq T\} \\ &\geq L_\psi(T) \cdot \frac{1}{n} \sum_{i=1}^n \varphi_{\tau'\|\Delta\|_2}(x_i^T\Delta) \cdot \gamma_T(x_i^T\beta), \end{aligned} \tag{98}$$

where the first equality is the expansion (93) and the second inequality uses the bound (97).

Now define the subset of $\mathbb{R}^p \times \mathbb{R}^p$ via

$$\mathbb{A}_\delta := \left\{ (\beta, \Delta) : \beta \in \mathbb{B}_2(3) \cap \mathbb{B}_1(R),\ \Delta \in \mathbb{B}_2(3),\ \frac{\|\Delta\|_1}{\|\Delta\|_2} \leq \delta \right\},$$

as well as the random variable

$$Z(\delta) := \sup_{(\beta,\Delta) \in \mathbb{A}_\delta} \frac{1}{\|\Delta\|_2^2} \left| \frac{1}{n} \sum_{i=1}^n \varphi_{\tau'\|\Delta\|_2}(x_i^T\Delta) \cdot \gamma_T(x_i^T\beta) - \mathbb{E}\left[\varphi_{\tau'\|\Delta\|_2}(x_i^T\Delta)\, \gamma_T(x_i^T\beta)\right] \right|.$$

For any pair $(\beta, \Delta) \in \mathbb{A}_\delta$, we have

$$\begin{aligned}
&\mathbb{E}\big[(x_i^T\Delta)^2 - \varphi_{\tau'\|\Delta\|_2}(x_i^T\Delta)\cdot\gamma_T(x_i^T\beta)\big] \\
&\qquad \le \mathbb{E}\left[(x_i^T\Delta)^2\mathbb{I}\left\{|x_i^T\Delta| \ge \frac{\tau'\|\Delta\|_2}{2}\right\}\right] + \mathbb{E}\left[(x_i^T\Delta)^2\mathbb{I}\left\{|x_i^T\beta| \ge \frac{T}{2}\right\}\right] \\
&\qquad \le \sqrt{\mathbb{E}\left[(x_i^T\Delta)^4\right]}\cdot\left(\sqrt{\mathbb{P}\left(|x_i^T\Delta| \ge \frac{\tau'\|\Delta\|_2}{2}\right)} + \sqrt{\mathbb{P}\left(|x_i^T\beta| \ge \frac{T}{2}\right)}\right) \\
&\qquad \le \sigma_x^2\|\Delta\|_2^2 \cdot c\exp\left(-\frac{c'\tau'^2}{\sigma_x^2}\right),
\end{aligned}$$

where we have used Cauchy-Schwarz and a tail bound for sub-Gaussians, assuming $\beta \in \mathbb{B}_2(3)$. It follows that for τ' chosen such that

$$c\sigma_x^2\exp\left(-\frac{c'\tau'^2}{\sigma_x^2}\right) = \frac{\lambda_{\min}\left(\mathbb{E}[x_i x_i^T]\right)}{2}, \tag{99}$$

we have the lower bound

$$\mathbb{E}\left[\varphi_{\tau'\|\Delta\|_2}(x_i^T\Delta)\cdot\gamma_T(x_i^T\beta)\right] \ge \frac{\lambda_{\min}\left(\mathbb{E}[x_i x_i^T]\right)}{2}\cdot\|\Delta\|_2^2. \tag{100}$$

By construction of φ, each summand in the expression for $Z(\delta)$ is sandwiched as

$$0 \le \frac{1}{\|\Delta\|_2^2}\cdot\varphi_{\tau'\|\Delta\|_2}(x_i^T\Delta)\cdot\gamma_T(x_i^T\beta) \le \frac{\tau'^2}{4}.$$

Consequently, applying the bounded differences inequality yields

$$\mathbb{P}\left(Z(\delta) \ge \mathbb{E}[Z(\delta)] + \frac{\lambda_{\min}\left(\mathbb{E}[x_i x_i^T]\right)}{4}\right) \le c_1\exp(-c_2 n). \tag{101}$$

Furthermore, by Lemmas 12 and 13 in Appendix E, we have

$$\mathbb{E}[Z(\delta)] \le 2\sqrt{\frac{\pi}{2}}\cdot\mathbb{E}\left[\sup_{(\beta,\Delta)\in\mathbb{A}_\delta}\frac{1}{\|\Delta\|_2^2}\left|\frac{1}{n}\sum_{i=1}^n g_i\Big(\varphi_{\tau'\|\Delta\|_2}(x_i^T\Delta)\cdot\gamma_T(x_i^T\beta)\Big)\right|\right], \tag{102}$$

where the g_i's are i.i.d. standard Gaussians. Conditioned on $\{x_i\}_{i=1}^n$, define the Gaussian processes

$$Z_{\beta,\Delta} := \frac{1}{\|\Delta\|_2^2}\cdot\frac{1}{n}\sum_{i=1}^n g_i\Big(\varphi_{\tau'\|\Delta\|_2}(x_i^T\Delta)\cdot\gamma_T(x_i^T\beta)\Big),$$

and note that for pairs (β, Δ) and $(\widetilde{\beta}, \widetilde{\Delta})$, we have

$$\operatorname{var}\left(Z_{\beta,\Delta} - Z_{\widetilde{\beta},\widetilde{\Delta}}\right) \le 2\operatorname{var}\left(Z_{\beta,\Delta} - Z_{\widetilde{\beta},\Delta}\right) + 2\operatorname{var}\left(Z_{\widetilde{\beta},\Delta} - Z_{\widetilde{\beta},\widetilde{\Delta}}\right),$$

with

$$\begin{aligned}\operatorname{var}\left(Z_{\beta,\Delta}-Z_{\widetilde{\beta},\Delta}\right) &= \frac{1}{\|\Delta\|_2^4}\cdot\frac{1}{n^2}\sum_{i=1}^n \varphi^2_{\tau'\|\Delta\|_2}(x_i^T\Delta)\cdot\left(\gamma_T(x_i^T\beta)-\gamma_T(x_i^T\widetilde{\beta})\right)^2\\ &\le \frac{1}{n^2}\sum_{i=1}^n \frac{\tau'^4}{16}\cdot\frac{4}{T^2}\left(x_i^T(\beta-\widetilde{\beta})\right)^2,\end{aligned}$$

since $\varphi_{\tau'\|\Delta\|_2}\le\frac{\tau'^2\|\Delta\|_2^2}{4}$ and γ_T is $\frac{2}{T}$-Lipschitz. Similarly, using the homogeneity property

$$\frac{1}{c^2}\cdot\varphi_{ct}(cu)=\varphi_t(u),\qquad \forall c>0,$$

and the fact that $\varphi_{\tau'\|\Delta\|_2}$ is $\tau'\|\Delta\|_2$-Lipschitz, we have

$$\begin{aligned}\operatorname{var}\left(Z_{\widetilde{\beta},\Delta}-Z_{\widetilde{\beta},\widetilde{\Delta}}\right) &\le \frac{1}{n^2}\sum_{i=1}^n\gamma_T^2(x_i^T\widetilde{\beta})\left(\frac{\varphi_{\tau'\|\Delta\|_2}(x_i^T\Delta)}{\|\Delta\|_2^2}-\frac{\varphi_{\tau'\|\widetilde{\Delta}\|_2}(x_i^T\widetilde{\Delta})}{\|\widetilde{\Delta}\|_2^2}\right)^2\\ &= \frac{1}{n^2}\sum_{i=1}^n\frac{\gamma_T^2(x_i^T\widetilde{\beta})}{\|\Delta\|_2^4}\left(\varphi_{\tau'\|\Delta\|_2}(x_i^T\Delta)-\varphi_{\tau'\|\Delta\|_2}\left(x_i^T\widetilde{\Delta}\cdot\frac{\|\Delta\|_2}{\|\widetilde{\Delta}\|_2}\right)\right)^2\\ &\le \frac{1}{n^2}\sum_{i=1}^n\frac{\tau'^2}{\|\Delta\|_2^2}\left(x_i^T\Delta-x_i^T\widetilde{\Delta}\cdot\frac{\|\Delta\|_2}{\|\widetilde{\Delta}\|_2}\right)^2\\ &= \frac{1}{n^2}\sum_{i=1}^n\tau'^2\left(\frac{x_i^T\Delta}{\|\Delta\|_2}-\frac{x_i^T\widetilde{\Delta}}{\|\widetilde{\Delta}\|_2}\right)^2.\end{aligned}$$

Defining the centered Gaussian process

$$Y_{\beta,\Delta}:=\frac{\tau'^2}{\sqrt{2}T}\cdot\frac{1}{n}\sum_{i=1}^n\widehat{g}_i\cdot x_i^T\beta+\frac{\sqrt{2}\tau'}{\|\Delta\|_2}\cdot\frac{1}{n}\sum_{i=1}^n\widetilde{g}_i\cdot x_i^T\Delta,$$

where the $\widehat{g}_i$'s and $\widetilde{g}_i$'s are independent standard Gaussians, it follows that

$$\operatorname{var}\left(Z_{\beta,\Delta}-Z_{\widetilde{\beta},\widetilde{\Delta}}\right)\le\operatorname{var}\left(Y_{\beta,\Delta}-Y_{\widetilde{\beta},\widetilde{\Delta}}\right).$$

Applying Lemma 14 in Appendix E, we then have

$$\mathbb{E}\left[\sup_{(\beta,\Delta)\in\mathbb{A}_\delta}Z_{\beta,\Delta}\right]\le 2\cdot\mathbb{E}\left[\sup_{(\beta,\Delta)\in\mathbb{A}_\delta}Y_{\beta,\Delta}\right].\tag{103}$$

Note further (cf. p.77 of Ledoux and Talagrand (1991)) that

$$\mathbb{E}\left[\sup_{(\beta,\Delta)\in\mathbb{A}_\delta}|Z_{\beta,\Delta}|\right]\le\mathbb{E}\left[|Z_{\beta_0,\Delta_0}|\right]+2\mathbb{E}\left[\sup_{(\beta,\Delta)\in\mathbb{A}_\delta}Z_{\beta,\Delta}\right],\tag{104}$$

for any $(\beta_0, \Delta_0) \in \mathbb{A}_\delta$, and furthermore,

$$\mathbb{E}\left[|Z_{\beta_0,\Delta_0}|\right] \leq \sqrt{\frac{2}{\pi}} \cdot \sqrt{\mathrm{var}\left(Z_{\beta_0,\Delta_0}\right)} \leq c_0 \cdot \sqrt{\frac{2}{\pi}} \cdot \sqrt{\frac{\tau'^2}{4n}}. \tag{105}$$

Finally,

$$\begin{aligned}\mathbb{E}\left[\sup_{(\beta,\Delta)\in\mathbb{A}_\delta} Y_{\beta,\Delta}\right] &\leq \frac{\tau'^2 R}{\sqrt{2}T} \cdot \mathbb{E}\left[\left\|\frac{1}{n}\sum_{i=1}^n \widehat{g}_i x_i\right\|_\infty\right] + \sqrt{2}\tau'\delta \cdot \mathbb{E}\left[\left\|\frac{1}{n}\sum_{i=1}^n \widetilde{g}_i x_i\right\|_\infty\right] \\ &\leq \frac{c\tau'^2 R\sigma_x}{T}\sqrt{\frac{\log p}{n}} + c\tau'\delta\sigma_x \cdot \sqrt{\frac{\log p}{n}}, \end{aligned} \tag{106}$$

by Lemma 16 in Appendix E. Combining (102), (103), (104), (105), and (106), we then obtain

$$\mathbb{E}[Z(\delta)] \leq \frac{c'\tau'^2 R\sigma_x}{T}\sqrt{\frac{\log p}{n}} + c'\tau'\delta\sigma_x \cdot \sqrt{\frac{\log p}{n}}. \tag{107}$$

Finally, combining (100), (101), and (107), we see that under the scaling $R\sqrt{\frac{\log p}{n}} \precsim 1$, we have

$$\begin{aligned}&\frac{1}{\|\Delta\|_2^2} \cdot \frac{1}{n}\sum_{i=1}^n \varphi_{\tau'\|\Delta\|_2}(x_i^T\Delta) \cdot \gamma_T(x_i^T\beta) \\ &\qquad \geq \frac{\lambda_{\min}\left(\mathbb{E}[x_i x_i^T]\right)}{4} - \left(\frac{c'\tau'^2 R\sigma_x}{T}\sqrt{\frac{\log p}{n}} + c'\tau'\delta\sigma_x\sqrt{\frac{\log p}{n}}\right) \\ &\qquad \geq \frac{\lambda_{\min}\left(\mathbb{E}[x_i x_i^T]\right)}{8} - c'\tau'\delta\sigma_x\sqrt{\frac{\log p}{n}}, \end{aligned} \tag{108}$$

uniformly over all $(\beta, \Delta) \in \mathbb{A}_\delta$, with probability at least $1 - c_1\exp(-c_2 n)$.

It remains to extend this bound to one that is uniform in the ratio $\frac{\|\Delta\|_1}{\|\Delta\|_2}$, which we do via a peeling argument (Alexander, 1987; van de Geer, 2000). Consider the inequality

$$\frac{1}{\|\Delta\|_2^2} \cdot \frac{1}{n}\sum_{i=1}^n \varphi_{\tau'\|\Delta\|_2}(x_i^T\Delta) \cdot \gamma_T(x_i^T\beta) \geq \frac{\lambda_{\min}\left(\mathbb{E}[x_i x_i^T]\right)}{8} - 2c'\tau'\sigma_x\frac{\|\Delta\|_1}{\|\Delta\|_2}\sqrt{\frac{\log p}{n}}, \tag{109}$$

as well as the event

$$\mathcal{E} := \left\{\text{Inequality (109) holds } \forall \|\beta\|_2 \leq 3 \text{ and } \tfrac{\|\Delta\|_1}{\|\Delta\|_2} \leq \tfrac{\lambda_{\min}\left(\mathbb{E}[x_i x_i^T]\right)}{16c'\tau\sigma_x}\sqrt{\tfrac{n}{\log p}}\right\}.$$

Define the function

$$f(\beta, \Delta; X) := \frac{\lambda_{\min}\left(\mathbb{E}[x_i x_i^T]\right)}{8} - \frac{1}{\|\Delta\|_2^2} \cdot \frac{1}{n}\sum_{i=1}^n \varphi_{\tau'\|\Delta\|_2}(x_i^T\Delta) \cdot \gamma_T(x_i^T\beta), \tag{110}$$

along with

$$g(\delta) := c'\tau'\sigma_x\delta\sqrt{\frac{\log p}{n}}, \qquad \text{and} \qquad h(\beta, \Delta) := \frac{\|\Delta\|_1}{\|\Delta\|_2}.$$

Note that (108) implies

$$\mathbb{P}\left(\sup_{h(\beta,\Delta)\leq\delta} f(\beta,\Delta;X)\geq g(\delta)\right)\leq c_1\exp(-c_2 n),\quad \text{for any } \delta>0, \tag{111}$$

where the sup is also restricted to $\{(\beta,\Delta):\beta\in\mathbb{B}_2(3)\cap\mathbb{B}_1(R),\ \Delta\in\mathbb{B}_2(3)\}$.

Since $\frac{\|\Delta\|_1}{\|\Delta\|_2}\geq 1$, we have

$$1\leq\ h(\beta,\Delta)\ \leq\frac{\lambda_{\min}\left(\mathbb{E}[x_i x_i^T]\right)}{16c'\tau'\sigma_x}\sqrt{\frac{n}{\log p}}, \tag{112}$$

over the region of interest. For each integer $m\geq 1$, define the set

$$\mathbb{V}_m:=\left\{(\beta,\Delta)\ \mid\ 2^{m-1}\mu\leq g(h(\beta,\Delta))\leq 2^m\mu\right\},$$

where $\mu=c'\tau'\sigma_x\sqrt{\frac{\log p}{n}}$. By a union bound, we then have

$$\mathbb{P}(\mathcal{E}^c)\leq\sum_{m=1}^{M}\mathbb{P}\left(\exists(\beta,\Delta)\in\mathbb{V}_m \text{ s.t. } f(\beta,\Delta;X)\geq 2g(h(\beta,\Delta))\right),$$

where the index m ranges up to $M:=\left\lceil\log\left(c\sqrt{\frac{n}{\log p}}\right)\right\rceil$ over the relevant region (112). By the definition (110) of f, we have

$$\mathbb{P}(\mathcal{E}^c)\leq\sum_{m=1}^{M}\mathbb{P}\left(\sup_{h(\beta,\Delta)\leq g^{-1}(2^m\mu)} f(\beta,\Delta;X)\geq 2^m\mu\right)\ \overset{(i)}{\leq}\ M\cdot c_1\exp(-c_2 n),$$

where inequality (i) applies the tail bound (111). It follows that

$$\mathbb{P}(\mathcal{E}^c)\leq c_1\exp\left(-c_2 n+\log\log\left(\frac{n}{\log p}\right)\right)\ \leq c_1'\exp\left(-c_2' n\right).$$

Multiplying through by $\|\Delta\|_2^2$ then yields the desired result.

Appendix E. Auxiliary Results

In this section, we provide some auxiliary results that are useful for our proofs. The first lemma concerns symmetrization and desymmetrization of empirical processes via Rademacher random variables:

Lemma 12 (Lemma 2.3.6 in van der Vaart and Wellner, 1996) *Let $\{Z_i\}_{i=1}^n$ be independent zero-mean stochastic processes. Then*

$$\frac{1}{2}\mathbb{E}\left[\sup_{t\in T}\left|\sum_{i=1}^n\epsilon_i Z_i(t_i)\right|\right]\leq\mathbb{E}\left[\sup_{t\in T}\left|\sum_{i=1}^n Z_i(t_i)\right|\right]\leq 2\mathbb{E}\left[\sup_{t\in T}\left|\sum_{i=1}^n\epsilon_i(Z_i(t_i)-\mu_i)\right|\right],$$

where the ϵ_i's are independent Rademacher variables and the functions $\mu_i:\mathcal{F}\to\mathbb{R}$ are arbitrary.

We also have a useful lemma that bounds the Gaussian complexity in terms of the Rademacher complexity:

Lemma 13 (Lemma 4.5 in Ledoux and Talagrand, 1991) *Let $Z_1, \ldots, Z_n$ be independent stochastic processes. Then*

$$\mathbb{E}\left[\sup_{t\in T}\left|\sum_{i=1}^n \epsilon_i Z_i(t_i)\right|\right] \leq \sqrt{\frac{\pi}{2}} \cdot \mathbb{E}\left[\sup_{t\in T}\left|\sum_{i=1}^n g_i Z_i(t_i)\right|\right],$$

where the ϵ_i's are Rademacher variables and the g_i's are standard normal.

We next state a version of the Sudakov-Fernique comparison inequality:

Lemma 14 (Corollary 3.14 in Ledoux and Talagrand, 1991) *Given a countable index set T, let $\{X(t), t \in T\}$ and $\{Y(t), t \in T\}$ be centered Gaussian processes such that*

$$\mathrm{var}\left(Y(s) - Y(t)\right) \leq \mathrm{var}\left(X(s) - X(t)\right), \qquad \forall (s,t) \in T \times T.$$

Then

$$\mathbb{E}\left[\sup_{t\in T} Y(t)\right] \leq 2 \cdot \mathbb{E}\left[\sup_{t\in T} X(t)\right].$$

A zero-mean random variable Z is sub-Gaussian with parameter σ if $\mathbb{P}(Z > t) \leq \exp(-\frac{t^2}{2\sigma^2})$ for all $t \geq 0$. The next lemma provides a standard bound on the expected maximum of N such variables (cf. Equation 3.6 in Ledoux and Talagrand, 1991):

Lemma 15 *Suppose $X_1, \ldots, X_N$ are zero-mean sub-Gaussian random variables such that $\max_{j=1,\ldots,N} \|X_j\|_{\psi_2} \leq \sigma$. Then $\mathbb{E}\left[\max_{j=1,\ldots,p} |X_j|\right] \leq c_0\, \sigma \sqrt{\log N}$, where $c_0 > 0$ is a universal constant.*

We also have a lemma about maxima of products of sub-Gaussian variables:

Lemma 16 *Suppose $\{g_i\}_{i=1}^n$ are i.i.d. standard Gaussians and $\{X_i\}_{i=1}^n \subseteq \mathbb{R}^p$ are i.i.d. sub-Gaussian vectors with parameter bounded by σ_x. Then as long as $n \geq c\sqrt{\log p}$ for some constant $c > 0$, we have*

$$\mathbb{E}\left[\left\|\frac{1}{n}\sum_{i=1}^n g_i X_i\right\|_\infty\right] \leq c' \sigma_x \sqrt{\frac{\log p}{n}}.$$

Proof Conditioned on $\{X_i\}_{i=1}^n$, for each $j = 1, \ldots, p$, the variable $\left|\frac{1}{n}\sum_{i=1}^n g_i X_{ij}\right|$ is zero-mean and sub-Gaussian with parameter bounded by $\frac{\sigma_x}{n}\sqrt{\sum_{i=1}^n X_{ij}^2}$. Hence, by Lemma 15, we have

$$\mathbb{E}\left[\left\|\frac{1}{n}\sum_{i=1}^n g_i X_i\right\|_\infty \middle| X\right] \leq \frac{c_0 \sigma_x}{n} \cdot \max_{j=1,\ldots,p} \sqrt{\sum_{i=1}^n X_{ij}^2} \cdot \sqrt{\log p},$$

implying that

$$\mathbb{E}\left[\left\|\frac{1}{n}\sum_{i=1}^{n} g_i X_i\right\|_\infty\right] \leq c_0\sigma_x\sqrt{\frac{\log p}{n}} \cdot \mathbb{E}\left[\max_j \sqrt{\frac{\sum_{i=1}^n X_{ij}^2}{n}}\right]. \tag{113}$$

Furthermore, $Z_j := \frac{\sum_{i=1}^n X_{ij}^2}{n}$ is an i.i.d. average of subexponential variables, each with parameter bounded by $c\sigma_x$. Since $\mathbb{E}[Z_j] \leq 2\sigma_x^2$, we have

$$\mathbb{P}\left(Z_j - \mathbb{E}[Z_j] \geq u + 2\sigma_x^2\right) \leq c_1 \exp\left(-\frac{c_2 n u}{\sigma_x}\right), \qquad \forall u \geq 0 \text{ and } 1 \leq j \leq p. \tag{114}$$

Now fix some $t \geq \sqrt{2\sigma_x^2}$. Since the $\{Z_j\}_{j=1}^p$ are all nonnegative, we have

$$\begin{aligned}
\mathbb{E}\left[\max_{j=1,\ldots,p} \sqrt{Z_j}\right] &\leq t + \int_t^\infty \mathbb{P}\left(\max_{j=1,\ldots,p} \sqrt{Z_j} > s\right) ds \\
&\leq t + \sum_{j=1}^p \int_t^\infty \mathbb{P}\left(\sqrt{Z_j} > s\right) ds \\
&\leq t + c_1 p \int_t^\infty \exp\left(-\frac{c_2 n (s^2 - 2\sigma_x^2)}{\sigma_x}\right) ds
\end{aligned}$$

where the final inequality follows from the bound (114) with $u = s^2 - 2\sigma_x^2$, valid as long as $s^2 \geq t^2 \geq 2\sigma_x^2$. Integrating, we have the bound

$$\mathbb{E}\left[\max_{j=1,\ldots,p} \sqrt{Z_j}\right] \leq t + c_1' p \sigma_x \exp\left(-\frac{c_2' n (t^2 - 2\sigma_x^2)}{\sigma_x^2}\right).$$

Since $n \gtrsim \sqrt{\log p}$ by assumption, setting t equal to a constant implies $\mathbb{E}\left[\max_j \sqrt{Z_j}\right] = \mathcal{O}(1)$, which combined with (113) gives the desired result. ■

Appendix F. Capped-ℓ_1 Penalty

In this section, we show how our results on nonconvex but subdifferentiable regularizers may be extended to include certain types of more complicated regularizers that do not possess (sub)gradients everywhere, such as the capped-ℓ_1 penalty.

In order to handle the case when ρ_λ has points where neither a gradient nor subderivative exists, we assume the existence of a function $\widetilde{\rho}_\lambda$ (possibly defined according to the particular local optimum $\widetilde{\beta}$ of interest), such that the following conditions hold:

Assumption 2

(i) The function $\widetilde{\rho}_\lambda$ is differentiable/subdifferentiable everywhere, and $\|\nabla\widetilde{\rho}_\lambda(\widetilde{\beta})\|_\infty \leq \lambda L$.

(ii) For all $\beta \in \mathbb{R}^p$, we have $\widetilde{\rho}_\lambda(\beta) \geq \rho_\lambda(\beta)$.

(iii) The equality $\widetilde{\rho}_\lambda(\widetilde{\beta}) = \rho_\lambda(\widetilde{\beta})$ holds.

(iv) There exists $\mu_1 \geq 0$ *such that* $\widetilde{\rho}_\lambda(\beta) + \frac{\mu_1}{2}\|\beta\|_2^2$ *is convex.*

(v) For some index set A *with* $|A| \leq k$ *and some parameter* $\mu_2 \geq 0$*, we have*

$$\widetilde{\rho}_\lambda(\beta^*) - \widetilde{\rho}_\lambda(\widetilde{\beta}) \leq \lambda L\|\widetilde{\beta}_A - \beta^*_A\|_1 - \lambda L\|\widetilde{\beta}_{A^c} - \beta^*_{A^c}\|_1 + \frac{\mu_2}{2}\|\widetilde{\beta} - \beta^*\|_2^2.$$

In addition, we assume conditions (i)–(iii) of Assumption 1 in Section 2.2 above.

When $\rho_\lambda(\beta) + \frac{\mu_1}{2}\|\beta\|_2^2$ is convex for some $\mu_1 \geq 0$ (as in the case of SCAD or MCP), we may take $\widetilde{\rho}_\lambda = \rho_\lambda$ and $\mu_2 = 0$ (cf. Lemma 5 in Appendix A.1). When no such convexification of ρ_λ exists (as in the case of the capped-ℓ_1 penalty), we instead construct a separate convex function $\widetilde{\rho}_\lambda$ to upper-bound ρ_λ and take $\mu_1 = 0$.

Under the conditions of Assumption 2, we have the following variant of Theorems 1 and 2:

Theorem 4 *Suppose* $\mathcal{L}_n$ *satisfies the RSC conditions* (4)*, and the functions* ρ_λ *and* $\widetilde{\rho}_\lambda$ *satisfy Assumption 1 and Assumption 2, respectively. Suppose* λ *is chosen according to the bound* (6) *and* $n \geq \frac{16R^2\max(\tau_1^2,\tau_2^2)}{\alpha_2^2}\log p$*. Then for any stationary point* $\widetilde{\beta}$ *of the program* (1)*, we have*

$$\|\widetilde{\beta} - \beta^*\|_2 \leq \frac{7\lambda L\sqrt{k}}{4\alpha_1 - 2\mu_1 - 2\mu_2}, \qquad \textit{and} \qquad \|\widetilde{\beta} - \beta^*\|_1 \leq \frac{28\lambda Lk}{2\alpha_1 - \mu_1 - \mu_2},$$

along with the prediction error bound

$$\langle \nabla\mathcal{L}_n(\widetilde{\beta}) - \nabla\mathcal{L}_n(\beta^*), \widetilde{\nu}\rangle \leq \lambda^2 L^2 k\left(\frac{21}{8\alpha_1 - 4\mu_1 - 4\mu_2)} + \frac{49(\mu_1 + \mu_2)}{8(2\alpha_1 - \mu_1 - \mu_2)^2}\right).$$

Proof

The proof is essentially the same as the proofs of Theorems 1 and 2, so we only mention a few key modifications here. First note that any local minimum $\widetilde{\beta}$ of the program (1) is a local minimum of $\mathcal{L}_n + \widetilde{\rho}_\lambda$, since

$$\mathcal{L}_n(\widetilde{\beta}) + \widetilde{\rho}_\lambda(\widetilde{\beta}) = \mathcal{L}_n(\widetilde{\beta}) + \rho_\lambda(\widetilde{\beta}) \leq \mathcal{L}_n(\beta) + \rho_\lambda(\beta) \leq \mathcal{L}_n(\beta) + \widetilde{\rho}_\lambda(\beta),$$

locally for all β in the constraint set, where the first inequality comes from the fact that $\widetilde{\beta}$ is a local minimum of $\mathcal{L}_n + \rho_\lambda$, and the second inequality holds because $\widetilde{\rho}_\lambda$ upper-bounds ρ_λ. Hence, the first-order condition (5) still holds with ρ_λ replaced by $\widetilde{\rho}_\lambda$. Consequently, (20) holds, as well.

Next, note that (22) holds as before, with ρ_λ replaced by $\widetilde{\rho}_\lambda$ and μ replaced by μ_1. By condition (v) on $\widetilde{\rho}_\lambda$, we then have (27) with μ replaced by $\mu_1 + \mu_2$. The remainder of the proof is essentially the same as before. Note that condition (v) does not include the extra factor of ξ appearing in Lemma 5, but this condition is actually strong enough for the proof arguments to hold, since we do not impose a positivity condition on the difference between $\|\widetilde{\nu}_A\|_1$ and $\|\widetilde{\nu}_{A^c}\|_1$. ■

Specializing now to the case of the capped-ℓ_1 penalty, we have the following lemma. For a fixed parameter $c \geq 1$, the capped-ℓ_1 penalty (Zhang and Zhang, 2012) is given by

$$\rho_\lambda(t) := \min\left\{\frac{\lambda^2 c}{2},\ \lambda|t|\right\}. \tag{115}$$

Lemma 17 *The capped-ℓ_1 regularizer* (115) *with parameter c satisfies the conditions of Assumption 2, with $\mu_1 = 0$, $\mu_2 = \frac{4}{c}$, and $L = 1$.*

Proof We will show how to construct an appropriate choice of $\widetilde{\rho}_\lambda$. Note that ρ_λ is piecewise linear and locally equal to $|t|$ in the range $\left[-\frac{\lambda c}{2}, \frac{\lambda c}{2}\right]$, and takes on a constant value outside that region. However, ρ_λ does not have either a gradient or subgradient at $t = \pm\frac{\lambda c}{2}$, hence is not "convexifiable" by adding a squared-ℓ_2 term.

For a fixed local optimum $\widetilde{\beta}$, we define the functions $\widetilde{\rho}_\lambda^j : \mathbb{R} \to \mathbb{R}$ via

$$\widetilde{\rho}_\lambda^j(t) = \begin{cases} \lambda|t|, & \text{if } |\widetilde{\beta}_j| \leq \frac{\lambda c}{2}, \\ \frac{\lambda^2 c}{2}, & \text{otherwise,} \end{cases}$$

and let $\widetilde{\rho}_\lambda(\beta) = \sum_{j=1}^p \widetilde{\rho}_\lambda^j(\beta_j)$, for $\beta \in \mathbb{R}^p$. Then

$$\widetilde{\rho}_\lambda(\beta) = \sum_{j \in T} \lambda|\beta_j| + \sum_{j \in T^c} \frac{\lambda^2 c}{2},$$

where $T := \left\{ j \mid |\widetilde{\beta}_j| \leq \frac{\lambda c}{2} \right\}$. It is easy to see that $\widetilde{\rho}_\lambda$ is a convex upper bound on ρ_λ, with $\widetilde{\rho}_\lambda(\widetilde{\beta}) = \rho_\lambda(\widetilde{\beta})$, since $\widetilde{\rho}_\lambda^j(\widetilde{\beta}_j) = \rho_\lambda(\widetilde{\beta}_j)$ for all j. Then

$$\widetilde{\rho}_\lambda(\beta^*) - \widetilde{\rho}_\lambda(\widetilde{\beta}) \;=\; \sum_{j \in S \cap T} \left(\widetilde{\rho}_\lambda^j(\beta_j^*) - \widetilde{\rho}_\lambda^j(\widetilde{\beta}_j) \right) + \sum_{j \in S^c \cap T} \left(\widetilde{\rho}_\lambda^j(\beta_j^*) - \widetilde{\rho}_\lambda^j(\widetilde{\beta}_j) \right), \tag{116}$$

using decomposability of $\widetilde{\rho}_\lambda$. Furthermore, $\widetilde{\rho}_\lambda^j(\beta_j^*) = \widetilde{\rho}_\lambda^j(0) = 0$ for $j \in S^c \cap T$, and for $j \in T$, we have

$$\widetilde{\rho}_\lambda^j(\beta_j^*) - \widetilde{\rho}_\lambda^j(\widetilde{\beta}_j) \leq \lambda|\beta_j^*| - \lambda|\widetilde{\beta}_j| \leq \lambda|\widetilde{\nu}_j|,$$

whereas for $j \notin T$, we have $\widetilde{\rho}_\lambda^j(\beta_j^*) - \widetilde{\rho}_\lambda^j(\widetilde{\beta}_j) = 0 \leq \lambda|\widetilde{\nu}_j|$. Combined with (116), we obtain

$$\begin{aligned} \widetilde{\rho}_\lambda(\beta^*) - \widetilde{\rho}_\lambda(\widetilde{\beta}) &\leq \sum_{j \in S \cap T} \lambda|\widetilde{\nu}_j| - \sum_{j \in S^c \cap T} \widetilde{\rho}_\lambda^j(\widetilde{\beta}_j) \\ &= \lambda\|\widetilde{\nu}_{S \cap T}\|_1 - \sum_{j \in S^c \cap T} \rho_\lambda(\widetilde{\beta}_j) \\ &= \lambda\|\widetilde{\nu}_{S \cap T}\|_1 - \lambda\|\widetilde{\nu}_{S^c \cap T}\|_1 + \sum_{j \in S^c \cap T} \left(\lambda|\widetilde{\beta}_j| - \rho_\lambda(\widetilde{\beta}_j) \right). \end{aligned} \tag{117}$$

Now observe that

$$\lambda|t| - \rho_\lambda(t) = \begin{cases} 0, & \text{if} \quad |t| \leq \frac{\lambda c}{2}, \\ \lambda|t| - \frac{\lambda^2 c}{2}, & \text{if} \quad |t| > \frac{\lambda c}{2}, \end{cases}$$

and moreover, the derivative of $\frac{t^2}{c}$ always exceeds λ for $|t| > \frac{\lambda c}{2}$. Consequently, we have $\lambda|t| - \rho_\lambda(t) \leq \frac{t^2}{c}$ for all $t \in \mathbb{R}$. Substituting this bound into (117) yields

$$\widetilde{\rho}_\lambda(\beta^*) - \widetilde{\rho}_\lambda(\widetilde{\beta}) \leq \lambda\|\widetilde{\nu}_{S \cap T}\|_1 - \lambda\|\widetilde{\nu}_{S^c \cap T}\|_1 + \frac{1}{c}\|\widetilde{\nu}_{S^c \cap T}\|_2^2 \leq \lambda\|\widetilde{\nu}_S\|_1 - \lambda\|\widetilde{\nu}_{S^c \cap T}\|_1 + \frac{1}{c}\|\widetilde{\nu}_{S^c \cap T}\|_2^2. \tag{118}$$

Finally, note that

$$\lambda\|\widetilde{\nu}_{S^c\cap T^c}\|_1 = \lambda\|\widetilde{\beta}_{S^c\cap T^c}\|_1 \leq \frac{2}{c}\|\widetilde{\beta}_{S^c\cap T^c}\|_2^2 = \frac{2}{c}\|\widetilde{\nu}_{S^c\cap T^c}\|_2^2, \tag{119}$$

since $\lambda|t| \leq \frac{2t^2}{c}$ when $|t| \geq \frac{\lambda c}{2}$. Combining (118) and (119) then yields

$$\widetilde{\rho}_\lambda(\beta^*) - \widetilde{\rho}_\lambda(\widetilde{\beta}) \leq \lambda\|\widetilde{\nu}_S\|_1 - \lambda\|\widetilde{\nu}_{S^c}\|_1 + \frac{2}{c}\|\widetilde{\nu}_{S^c}\|_2^2 \leq \lambda\|\widetilde{\nu}_S\|_1 - \lambda\|\widetilde{\nu}_{S^c}\|_1 + \frac{2}{c}\|\widetilde{\nu}\|_2^2,$$

which is condition (v) of Assumption 2 on $\widetilde{\rho}_\lambda$ with $L = 1$, $A = S$, and $\mu_2 = \frac{4}{c}$. The remaining conditions are easy to verify (see also Zhang and Zhang, 2012). ■

References

A. Agarwal, S. Negahban, and M. J. Wainwright. Fast global convergence of gradient methods for high-dimensional statistical recovery. *Annals of Statistics*, 40(5):2452–2482, 2012.

K. S. Alexander. Rates of growth and sample moduli for weighted empirical processes indexed by sets. *Probability Theory and Related Fields*, 75:379–423, 1987.

O. Banerjee, L. El Ghaoui, and A. d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *Journal of Machine Learning Research*, 9:485–516, 2008.

D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, 1999.

P. J. Bickel, Y. Ritov, and A. Tsybakov. Simultaneous analysis of Lasso and Dantzig selector. *Annals of Statistics*, 37(4):1705–1732, 2009.

P. Breheny and J. Huang. Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. *Annals of Applied Statistics*, 5(1):232–253, 2011.

R. J. Carroll, D. Ruppert, and L. A. Stefanski. *Measurement Error in Nonlinear Models*. Chapman and Hall, 1995.

L. Chen and Y. Gu. The convergence guarantees of a non-convex approach for sparse recovery. *IEEE Transactions on Signal Processing*, 62(15):3754–3767, 2014.

J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96:1348–1360, 2001.

J. Fan, Y. Feng, and Y. Wu. Network exploration via the adaptive LASSO and SCAD penalties. *Annals of Applied Statistics*, pages 521–541, 2009.

J. Fan, L. Xue, and H. Zou. Strong oracle optimality of folded concave penalized estimation. *The Annals of Statistics*, 42(3):819–849, 06 2014.

J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical Lasso. *Biostatistics*, 9(3):432–441, July 2008.

R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1990.

D. R. Hunter and R. Li. Variable selection using MM algorithms. *Annals of Statistics*, 33 (4):1617–1642, 2005.

M. Ledoux and M. Talagrand. *Probability in Banach Spaces: Isoperimetry and Processes*. Springer-Verlag, New York, NY, 1991.

E. L. Lehmann and G. Casella. *Theory of Point Estimation*. Springer Verlag, 1998.

P. Loh and M. J. Wainwright. High-dimensional regression with noisy and missing data: Provable guarantees with non-convexity. *Annals of Statistics*, 40(3):1637–1664, 2012.

P. Loh and M. J. Wainwright. Structure estimation for discrete graphical models: Generalized covariance matrices and their inverses. *The Annals of Statistics*, 41(6):3022–3049, 12 2013a.

P. Loh and M. J. Wainwright. Regularized M-estimators with nonconvexity: Statistical and algorithmic theory for local optima. *arXiv e-prints*, May 2013b. Available at `http://arxiv.org/abs/1305.2436`.

P. Loh and M. J. Wainwright. Regularized M-estimators with nonconvexity: Statistical and algorithmic theory for local optima. In *NIPS*, pages 476–484, 2013c.

R. Mazumder, J. H. Friedman, and T. Hastie. Sparsenet: Coordinate descent with nonconvex penalties. *Journal of the American Statistical Association*, 106(495):1125–1138, 2011.

P. McCullagh and J. A. Nelder. *Generalized Linear Models (Second Edition)*. London: Chapman & Hall, 1989.

S. Negahban, P. Ravikumar, M. J. Wainwright, and B. Yu. A unified framework for high-dimensional analysis of M-estimators with decomposable regularizers. *Statistical Science*, 27(4):538–557, December 2012. See arXiv version for lemma/propositions cited here.

Y. Nesterov. Gradient methods for minimizing composite objective function. CORE Discussion Papers 2007076, Universit Catholique de Louvain, Center for Operations Research and Econometrics (CORE), 2007. URL `http://EconPapers.repec.org/RePEc:cor:louvco:2007076`.

Y. Nesterov and A. Nemirovskii. *Interior Point Polynomial Algorithms in Convex Programming*. SIAM studies in applied and numerical mathematics. Society for Industrial and Applied Mathematics, 1987.

G. Raskutti, M. J. Wainwright, and B. Yu. Minimax rates of estimation for high-dimensional linear regression over ℓ_q-balls. *IEEE Transactions on Information Theory*, 57(10):6976–6994, 2011.

P. Ravikumar, M. J. Wainwright, G. Raskutti, and B. Yu. High-dimensional covariance estimation by minimizing ℓ_1-penalized log-determinant divergence. *Electronic Journal of Statistics*, 4:935–980, 2011.

M. Rosenbaum and A. B. Tsybakov. Sparse recovery under matrix uncertainty. *Annals of Statistics*, 38:2620–2651, 2010.

A. J. Rothman, P. J. Bickel, E. Levina, and J. Zhu. Sparse permutation invariant covariance estimation. *Electronic Journal of Statistics*, 2:494–515, 2008.

S. van de Geer. *Empirical Processes in M-Estimation*. Cambridge University Press, 2000.

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

S. A. Vavasis. Complexity issues in global optimization: A survey. In *Handbook of Global Optimization*, pages 27–41. Kluwer, 1995.

J.-P. Vial. Strong convexity of sets and functions. *Journal of Mathematical Economics*, 9 (1-2):187–205, January 1982.

M. J. Wainwright. Structured regularizers for high-dimensional problems: Statistical and computational issues. *Annual Review of Statistics and Its Application*, 1(1):233–253, 2014.

Z. Wang, H. Liu, and T. Zhang. Optimal computational and statistical rates of convergence for sparse nonconvex learning problems. *The Annals of Statistics*, 42(6):2164–2201, 12 2014.

M. Yuan and Y. Lin. Model selection and estimation in the Gaussian graphical model. *Biometrika*, 94(1):19–35, 2007.

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

C.-H. Zhang and T. Zhang. A general theory of concave regularization for high-dimensional sparse estimation problems. *Statistical Science*, 27(4):576–593, 2012.

H. Zou and R. Li. One-step sparse estimates in nonconcave penalized likelihood models. *Annals of Statistics*, 36(4):1509–1533, 2008.

Journal of Machine Learning Research 16 (2015) 617-652 Submitted 11/12; Revised 9/14; Published 3/15

Generalized Hierarchical Kernel Learning

Pratik Jawanpuria PRATIK.J@CSE.IITB.AC.IN
Jagarlapudi Saketha Nath SAKETH@CSE.IITB.AC.IN
Ganesh Ramakrishnan GANESH@CSE.IITB.AC.IN
Department of Computer Science and Engineering
Indian Institute of Technology Bombay
Mumbai 400076, INDIA

Editor: Francis Bach

Abstract

This paper generalizes the framework of Hierarchical Kernel Learning (HKL) and illustrates its utility in the domain of rule learning. HKL involves Multiple Kernel Learning over a set of given base kernels assumed to be embedded on a directed acyclic graph. This paper proposes a two-fold generalization of HKL: the first is employing a generic ℓ_1/ℓ_ρ block-norm regularizer ($\rho \in (1, 2]$) that alleviates a key limitation of the HKL formulation. The second is a generalization to the case of multi-class, multi-label and more generally, multi-task applications. The main technical contribution of this work is the derivation of a highly specialized partial dual of the proposed generalized HKL formulation and an efficient mirror descent based active set algorithm for solving it. Importantly, the generic regularizer enables the proposed formulation to be employed in the Rule Ensemble Learning (REL) where the goal is to construct an ensemble of conjunctive propositional rules. Experiments on benchmark REL data sets illustrate the efficacy of the proposed generalizations.

Keywords: multiple kernel learning, mixed-norm regularization, multi-task learning, rule ensemble learning, active set method

1. Introduction

A Multiple Kernel Learning (MKL) (Lanckriet et al., 2004; Bach et al., 2004) framework for construction of sparse linear combinations of base kernels embedded on a directed acyclic graph (DAG) was recently proposed by Bach (2008). Since the DAG induces hierarchical relations between the base kernels, this framework is more commonly known as Hierarchical Kernel Learning (HKL). It has been established that HKL provides a powerful algorithm for task specific non-linear feature selection. HKL employs a carefully designed ℓ_1/ℓ_2 block-norm regularizer: ℓ_1-norm across some predefined components associated with the DAG and ℓ_2-norm within each such component. However, the sparsity pattern of kernel (feature) selection induced by this regularizer is somewhat restricted: *a kernel is selected only if the kernels associated with all its ancestors in the DAG are selected.* In addition, it can be proved that the weight of the kernel associated with a (selected) node will always be greater than the weight of the kernels associated with its descendants. Such a restricted selection pattern and weight bias may limit the applicability of HKL in real world problems.

This paper proposes a two-fold generalization of HKL. The first is employing a $\ell_1/\ell_\rho, \rho \in (1, 2)$, block-norm regularizer that mitigates the above discussed weight and selection bias

among the kernels, henceforth termed as gHKL. Note that for the special case of $\rho = 2$, gHKL renders the HKL regularizer. Further, gHKL is generalized to the paradigm of Multi-task Learning (MTL), where multiple related tasks need to be learnt jointly. We consider the MTL setup where the given learning tasks share a common sparse feature space (Lounici et al., 2009; Jawanpuria and Nath, 2011; Obozinski et al., 2011). Our goal is to construct a shared sparse feature representation that is suitable for all the given related tasks. We pose the problem of learning this shared feature space as that of learning a shared kernel, common across all the tasks. The proposed generalization is henceforth referred to as gHKL$_{\rm MT}$. In addition to learning a common feature representation, gHKL$_{\rm MT}$ is generic enough to model additional correlations existing among the given tasks.

Though employing a $\ell_1/\ell_\rho, \rho \in (1, 2)$, regularizer is an incremental modification to the HKL formulation, devising an algorithm for solving it is not straightforward. The projected gradient descent employed in the active set algorithm for solving HKL (Bach, 2008) can no longer be employed for solving gHKL as projections onto ℓ_ρ-norm balls are known to be significantly more challenging than those onto ℓ_1-norm balls (Liu and Ye, 2010). Hence naive extensions of the existing HKL algorithm will not scale well. Further, the computational challenge is compounded with the generalization for learning multiple tasks jointly. The key technical contribution of this work is the derivation of a highly specialized partial dual of the gHKL/gHKL$_{\rm MT}$ formulations and an efficient mirror descent (Ben-Tal and Nemirovski, 2001; Beck and Teboulle, 2003) based active set algorithm for solving it. The dual presented here is an elegant convex optimization problem with a Lipschitz continuous objective and constrained over a simplex. Moreover, the gradient of the objective can be obtained by solving a known and well-studied variant of the MKL formulation. This motivates employing the mirror descent algorithm that is known to solve such problems efficiently. Further efficiency is brought in by employing an active set method similar in spirit to that in Bach (2008).

A significant portion of this paper focuses on the application of Rule Ensemble Learning (REL) (Dembczyński et al., 2010, 2008), where HKL has not been previously explored. Given a set of basic propositional features describing the data, the goal in REL is to construct a compact ensemble of conjunctions with the given propositional features that generalizes well for the problem at hand. Such ensembles are expected to achieve a good trade-off between interpretability and generalization ability. REL approaches (Cohen and Singer, 1999; Friedman and Popescu, 2008; Dembczyński et al., 2010) have additionally addressed the problem of learning a compact set of rules that generalize well in order to maintain their readability. One way to construct a compact ensemble is to consider a linear model involving all possible conjunctions of the basic propositional features and then performing a ℓ_1-norm regularized empirical risk minimization (Friedman and Popescu, 2008; Dembczyński et al., 2010). Since this is a computationally infeasible problem, even with moderate number of basic propositions, the existing methods either approximate such a regularized solution using strategies such as shrinkage (Friedman and Popescu, 2008; Dembczyński et al., 2010, 2008) or resort to post-pruning (Cohen and Singer, 1999). This work proposes to solve a variant of this regularized empirical risk minimization problem optimally using the framework of gHKL. The key idea is to define kernels representing every possible conjunction and arranging them on a DAG. The proposed gHKL regularizer is applied on this DAG of kernels, leading to a sparse combination of promising conjunctions. Note that

with such a setup, the size of the gHKL optimization problem is exponential in the number of basic propositional features. However, a key result in the paper shows that the proposed gHKL algorithm is guaranteed to solve this exponentially large problem with a complexity polynomial in the final active set[1] size. Simulations on benchmark binary (and multiclass) classification data sets show that gHKL (and gHKL_{MT}) indeed constructs a compact ensemble that on several occasions outperforms state-of-the-art REL algorithms in terms of generalization ability. These results also illustrate the benefits of the proposed generalizations over HKL: i) the ensembles constructed with gHKL (with low ρ values) involve fewer number of rules than with HKL; though the accuracies are comparable ii) gHKL_{MT} can learn rule ensemble on multiclass problems; whereas HKL is limited to two-class problems.

The rest of the paper[2] is organized as follows. Section 2 introduces the classical Multiple Kernel Learning setup, briefly reviews the HKL framework and summarizes the existing works in Multi-task Learning. In Section 3, we present the proposed gHKL and gHKL_{MT} formulations. The key technical derivation of the specialized dual is also presented in this section. The proposed mirror descent based active set algorithm for solving $\text{gHKL}/\text{gHKL}_{\text{MT}}$ formulations is discussed in Section 4. In Section 5, we propose to solve the REL problem by employing the gHKL formulation and discuss its details. In Section 6, we report empirical evaluations of gHKL and gHKL_{MT} formulations for REL on benchmark binary and multiclass data sets respectively. Section 7 concludes the paper.

2. Related Works

This section provides a brief introduction to the Multiple Kernel Learning (MKL) framework, the HKL setup and formulation (Bach, 2008, 2009) as well as the existing works in Multi-task Learning.

2.1 Multiple Kernel Learning Framework

We begin by discussing the regularized risk minimization framework (Vapnik, 1998), which has been employed in the proposed formulations.

Consider a learning problem like classification or regression and let its training data be denoted by $\mathcal{D} = \{(\mathbf{x}_i, y_i), i = 1, \ldots, m \mid \mathbf{x}_i \in \mathcal{X}, y_i \in \mathbb{R}\ \forall i\}$, where $(\mathbf{x}_i, y_i)$ represents the i^{th} input-output pair. The aim is to learn an affine prediction function $F(\mathbf{x})$ that generalize well on unseen data. Given a positive definite kernel k that induces a feature map $\phi_k(\cdot)$, the prediction function can be written as: $F(\mathbf{x}) = \langle f, \phi_k(\mathbf{x})\rangle_{\mathcal{H}_k} - b$. Here $\mathcal{H}_k$ is the Reproducing Kernel Hilbert Space (RKHS) (Schölkopf and Smola, 2002) associated with the kernel k, endowed with an inner product $\langle\cdot,\cdot\rangle_{\mathcal{H}_k}$, and $f \in \mathcal{H}_k, b \in \mathbb{R}$ are the model parameters to be learnt. A popular framework to learn these model parameters is the regularized risk minimization (Vapnik, 1998), which considers the following problem:

$$\min_{f \in \mathcal{H}_k, b \in \mathbb{R}} \frac{1}{2}\Omega(f)^2 + C\sum_{i=1}^{m} \ell(y_i, F(\mathbf{x}_i)), \tag{1}$$

1. Roughly, this is the number of selected conjunctions and is potentially far less than the total number of conjunctions.
2. Preliminary results of this work were reported in Jawanpuria et al. (2011).

where $\Omega(\cdot)$ is a norm based regularizer, $\ell : \mathbb{R}\times\mathbb{R} \to \mathbb{R}$ is a suitable convex loss function and C is a regularization parameter. As an example, the support vector machine (SVM) (Vapnik, 1998) employs $\Omega(f) = \|f\|_{\mathcal{H}_k}$. From the *representer theorem* (Schölkopf and Smola, 2002), we know that the optimal f has the following form $f(\cdot) = \sum_i^m \alpha_i k(\cdot, \mathbf{x}_i)$ where $\alpha = (\alpha_i)_{i=1}^m$ is a vector of coefficients to be learnt.

It can be observed from above that the kernel definition plays a crucial role in defining the quality of the solution obtained by solving (1). Hence learning a kernel suitable to the problem at hand has been an active area of research over the past few years. One way to learn kernels is via the Multiple Kernel Learning (MKL) framework (Lanckriet et al., 2004; Bach et al., 2004). Lanckriet et al. (2004) proposed to learn the kernel k as a conic combination of the given base kernels $k_1, \ldots, k_l$: $k = \sum_{i=1}^l \eta_i k_i,\ \eta_i \geq 0\ \forall\ i$. Here $\eta = (\eta_i)_{i=1}^l$ is a coefficient vector to be (additionally) learnt in the optimization problem (1). In this setting, the feature map with respect to the kernel k is given by $\phi_k = (\sqrt{\eta_i}\phi_{k_i})_{i=1}^l$ (see Rakotomamonjy et al., 2008, for details). It is a weighted concatenation of feature maps induced by the individual base kernels. Hence, sparse kernel weights will result in a low dimensional ϕ_k. Some of the additional constraints on η explored in the existing MKL works are ℓ_1-norm constraint (Bach et al., 2004; Rakotomamonjy et al., 2008), ℓ_p-norm constraint ($p > 1$) (Kloft et al., 2011; Vishwanathan et al., 2010; Aflalo et al., 2011), etc.

2.2 Hierarchical Kernel Learning

Hierarchical Kernel Learning (HKL) (Bach, 2008) is a generalization of MKL and assumes a hierarchy over the given base kernels. The base kernels are embedded on a DAG and a carefully designed ℓ_1/ℓ_2 block-norm regularization over the associated RKHS is proposed to induce a specific sparsity pattern over the selected base kernels. We begin by discussing its kernel setup.

Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be the given DAG with $\mathcal{V}$ denoting the set of vertices and $\mathcal{E}$ denoting the set of edges. The DAG structure entails relationships like parent, child, ancestor and descendant (Cormen et al., 2009). Let $D(v)$ and $A(v)$ represent the set of descendants and ancestors of the node v in the $\mathcal{G}$. It is assumed that both $D(v)$ and $A(v)$ include the node v. For any subset of nodes $\mathcal{W} \subset \mathcal{V}$, the *hull* and *sources* of $\mathcal{W}$ are defined as:

$$hull(\mathcal{W}) = \bigcup_{w\in\mathcal{W}} A(w), \quad sources(\mathcal{W}) = \{w \in \mathcal{W} \mid A(w) \cap \mathcal{W} = \{w\}\}.$$

The size and complement of $\mathcal{W}$ are denoted by $|\mathcal{W}|$ and $\mathcal{W}^c$ respectively. Let $k_v : \mathcal{X}\times\mathcal{X} \to \mathbb{R}$ be the *positive definite* kernel associated with the vertex $v \in \mathcal{V}$. In addition, let $\mathcal{H}_{k_v}$ be its associated RKHS and ϕ_{k_v} be its induced feature map. Given this, HKL employs the following prediction function:

$$F(\mathbf{x}) = \sum_{v\in\mathcal{V}} \langle f_v, \phi_{k_v}(\mathbf{x})\rangle_{\mathcal{H}_{k_v}} - b,$$

which is an affine model parameterized by $f = (f_v)_{v\in\mathcal{V}}$, the tuple with entries as $f_v \in \mathcal{H}_{k_v}$ and $b \in \mathbb{R}$. Some more notations follow: for any subset of nodes $\mathcal{W} \subset \mathcal{V}$, $f_\mathcal{W} = (f_v)_{v\in\mathcal{W}}$ and $\phi_\mathcal{W} = (\phi_v)_{v\in\mathcal{W}}$. In general, the entries in a vector are referred to using an appropriate subscript, i.e., entries in $\mathbf{u} \in \mathbb{R}^d$ are denoted by $u_1, \ldots, u_d$. The kernels are denoted by the lower case 'k' and the corresponding Gram matrices are denoted by the upper case 'K'.

HKL formulates the problem of learning the optimal prediction function F as the following regularized risk minimization problem:

$$\min_{f_v \in \mathcal{H}_{k_v} \forall v \in \mathcal{V}, b \in \mathbb{R}} \ \frac{1}{2} \left(\sum_{v \in \mathcal{V}} d_v \|f_{D(v)}\|_2 \right)^2 + C \sum_{i=1}^{m} \ell\left(y_i, F(\mathbf{x}_i)\right), \tag{2}$$

where $\|f_{D(v)}\|_2 = \left(\sum_{w \in D(v)} \|f_w\|^2 \right)^{\frac{1}{2}} \ \forall v \in \mathcal{V}$, $\ell(\cdot, \cdot)$ is a suitable convex loss function and $(d_v)_{v \in \mathcal{V}}$ are given non-negative parameters.

As is clear from (2), HKL employs a ℓ_1/ℓ_2 block-norm regularizer, which is known to promote group sparsity (Yuan and Lin, 2006). Its implications are discussed in the following. For most of $v \in \mathcal{V}$, $\|f_{D(v)}\|_2 = 0$ at optimality due to the sparsity inducing nature of the ℓ_1-norm. Moreover $(\|f_{D(v)}\|_2 = 0) \Rightarrow (f_w = 0 \ \forall w \in D(v))$. Thus it is expected that most f_v will be zero at optimality. This implies that the prediction function involves very few kernels. Under mild conditions on the kernels (being strictly positive), it can be shown that this hierarchical penalization induces the following sparsity pattern: $(f_w \neq 0) \Rightarrow (f_v \neq 0 \ \forall v \in A(w))$. In other words, if the prediction function employs a kernel k_w then it *certainly* employs *all* the kernels associated with the ancestor nodes of w.

Bach (2008) proposes to solve the following equivalent variational formulation:

$$\min_{\gamma \in \Delta_1} \ \min_{f_v \in \mathcal{H}_{k_v} \forall v \in \mathcal{V}, b \in \mathbb{R}} \ \frac{1}{2} \sum_{w \in \mathcal{V}} \delta_w(\gamma)^{-1} \|f_w\|^2 + C \sum_{i=1}^{m} \ell\left(y_i, F(\mathbf{x}_i)\right), \tag{3}$$

where $\Delta_1 = \left\{ \mathbf{z} \in \mathbb{R}^{|\mathcal{V}|} \mid \mathbf{z} \geq 0, \sum_{v \in \mathcal{V}} \mathbf{z}_v \leq 1 \right\}$ and $\delta_w(\gamma)^{-1} = \sum_{v \in A(w)} \frac{d_v^2}{\gamma_v}$. From the representer theorem (Schölkopf and Smola, 2002), it follows that the effective kernel employed in the HKL is: $k = \sum_{w \in \mathcal{V}} \delta_w(\gamma) k_w$. Since the optimization problem (3) has a ℓ_1-norm constraint over γ variables, most γ_v at optimality are expected to be zero. Moreover the kernel weight $\delta_w(\gamma)$ is zero whenever $\gamma_v = 0$ for any $v \in A(w)$. Thus, the HKL performs a sparse selection of the base kernels and can be understood as a generalization of the classical MKL framework. However, the sparsity pattern for the kernels has the following restriction: if a kernel is not selected then none of the kernels associated with its descendants are selected, as $(\gamma_v = 0) \Rightarrow (\delta_w(\gamma) = 0 \ \forall w \in D(v))$. For the case of strictly positive kernels, it follows that a kernel is selected only if all the kernels associated with its ancestors are selected. In addition, the following relationship holds among the kernels weights: $\delta_v(\gamma) \geq \delta_w(\gamma) \ \forall w \in D(v)$ (strict inequality holds if $\delta_w(\gamma) > 0$). Hence, the weight of the kernel associated with a (selected) node is always be greater than the weight of the kernels associated with its descendants.

Since the size of γ is same as that of $\mathcal{V}$ and since the optimal γ is known to be sparse, Bach (2008) proposes an active set based algorithm (Lee et al., 2007) for solving (3). At each iteration of the active set algorithm, (3) is solved with respect to only those variables in the active set via the projected gradient descent technique (Rakotomamonjy et al., 2008).

As illustrated in Bach (2008), the key advantage of HKL is in performing non-linear feature selection. For example, consider the case where the input space is $\mathcal{X} = \mathbb{R}^n$ and let I be power set of $\{1, \ldots, n\}$. Consider the following 2^n kernels arranged on the usual subset lattice: $k_i(\mathbf{x}, \mathbf{x}') = \Pi_{j \in i} x_j x'_j \ \forall i \in I$. HKL can be applied in this setup to select

the promising sub-products of the input features over all possible sub-products. Please refer to Bach (2008) for more such pragmatic examples of kernels and corresponding DAGs. The most interesting result in Bach (2008) is that in all these examples where the size of the DAG is exponentially large, the computational complexity of the active set algorithm is polynomial in the training set dimensions and the active set size. Importantly, the complexity is independent of $|\mathcal{V}|$!

Though encouraging, the above discussed weight bias (in favor of the kernels towards the top of the DAG) and restricted kernel selection pattern may limit the applicability of HKL in real world problems. For instance, in case of the sub-product kernel example mentioned above, the following is true: a sub-product is selected only if all the products including it are selected. This clearly may lead to selection of many redundant sub-products (features). In Section 3, we present the proposed generalization that provides a more flexible kernel selection pattern by employing a $\ell_1/\ell_\rho,\ \rho \in (1,2)$, regularizer. A key result of this paper (refer Corollary 6) is that for all the cases discussed in Bach (2008), the proposed mirror descent based active set algorithm for solving the generalization has a computational complexity that is still polynomial in the training set dimensions and the active set size. In other words, the proposed generalization does not adversely affect the computational feasibility of the problem and hence is an interesting result in itself.

2.3 Multi-task Learning

Multi-task Learning (Caruana, 1997; Baxter, 2000) focuses on learning several prediction tasks simultaneously. This is in contrast with the usual approach of learning each task separately and independently. The key underlying idea behind MTL is that an appropriate sharing of information while learning *related* tasks will help in obtaining better prediction models. Various definitions of task-relatedness have been explored over the past few years like proximity of task parameters (Baxter, 2000; Evgeniou and Pontil, 2004; Xue et al., 2007; Jacob et al., 2008; Jawanpuria and Nath, 2012) or sharing common feature space (Ando and Zhang, 2005; Ben-David and Schuller, 2008; Argyriou et al., 2008; Lounici et al., 2009; Obozinski et al., 2011). Many learning settings like multiclass classification, multi-label classification or learning vector-valued function may be viewed as a special case of multi-task learning.

In this work, we consider the common setting in which the task parameters share a simultaneously sparse structure: only a small number of input features are relevant for each of the tasks and the set of such relevant features is common across all the tasks (Turlach et al., 2005; Lounici et al., 2009). Existing works in this setting typically employ a group lasso penalty on the tasks parameters: ℓ_1/ℓ_2 block-norm (Lounici et al., 2009; Obozinski et al., 2011) or the ℓ_1/ℓ_∞ block-norm (Turlach et al., 2005; Negahban and Wainwright, 2009). Thus, they propose a multi-task regularizer of the form: $\Omega(f_1,\ldots,f_T) = \sum_{i=1}^{d}\left(\sum_{t=1}^{T}|f_{ti}|^q\right)^{\frac{1}{q}}$ where the input feature space is assumed to be d dimensional, f_t is the task parameter of the t^{th} task and $f_t = (f_{ti})_{i=1,\ldots,d}$ and $q = \{2,\infty\}$. Note that in addition to (sparse) shared feature selection, the ℓ_1/ℓ_∞ block-norm penalty also promote proximity among the task parameters.

We pose the problem of learning the shared features as that of learning a shared kernel, whose induced feature space is common across all the tasks. The shared kernel is

constructed as a sparse combination of the given base kernels. A hierarchical relationship exists over the given kernels (feature spaces). We employ a graph based ℓ_1/ℓ_ρ block-norm regularization over the task parameters that enable non-linear feature selection for multiple tasks simultaneously. The details of the proposed MTL formulation are discussed in the following section.

3. Generalized Hierarchical Kernel Learning

In this section, we present the proposed generalizations over HKL. As discussed earlier, the first generalization aims at mitigating the weight bias problem as well as the restrictions imposed on the kernel selection pattern of HKL, and is termed as gHKL. The gHKL formulation is then further generalized to the paradigm of MTL, the proposed formulation being termed as gHKL$_{\text{MT}}$. We begin by introducing the gHKL formulation.

3.1 gHKL Primal Formulation

Recall that HKL employs a ℓ_1/ℓ_2 block norm regularizer. As we shall understand in more detail later, a key reason for the kernel weight bias problem and the restricted sparsity pattern in HKL is the ℓ_2-norm regularization. One way to mitigate these restrictions is by employing the following generic regularizer:

$$\Omega_S(f) = \sum_{v \in \mathcal{V}} d_v \|f_{D(v)}\|_\rho, \tag{4}$$

where $f = (f_v)_{v \in \mathcal{V}}$, $\|f_{D(v)}\|_\rho = \left(\sum_{w \in D(v)} \|f_w\|^\rho\right)^{\frac{1}{\rho}}$ and $\rho \in (1, 2]$. The implications of the ℓ_1/ℓ_ρ block-norm regularization are discussed in the following. Since the ℓ_1-norm promotes sparsity, it follows that $\|f_{D(v)}\|_\rho = 0$ (that is $f_w = 0\ \forall w \in D(v)$) for most $v \in \mathcal{V}$. This phenomenon is similar as in HKL. But now, even in cases where $\|f_{D(v)}\|_\rho$ is not forced to zero by the ℓ_1-norm, many components of $f_{D(v)}$ tend to zero[3] (that is $f_w \to \mathbf{0}$ for many $w \in D(v)$) as the value of ρ tends to unity. Also note that $\rho = 2$ renders the HKL regularizer. To summarize, the proposed gHKL formulation is

$$\min_{f_v \in \mathcal{H}_{k_v} \forall v \in \mathcal{V}, b \in \mathbb{R}} \ \frac{1}{2}(\Omega_S(f))^2 + C \sum_{i=1}^{m} \ell\left(y_i, F(\mathbf{x}_i)\right). \tag{5}$$

We next present the gHKL$_{\text{MT}}$ formulation, which further generalizes gHKL to MTL paradigm.

3.2 gHKL$_{\text{MT}}$ Primal Formulation

We begin by introducing some notations for the multi-task learning setup. Let T be the number of tasks and let the training data for the t^{th} task be denoted by $\mathcal{D}_t = \{(\mathbf{x}_{ti}, y_{ti}),\ i = 1, \ldots, m \mid \mathbf{x}_{ti} \in \mathcal{X}, y_{ti} \in \mathbb{R}\ \forall i\}$, where $(\mathbf{x}_{ti}, y_{ti})$ represents the i^{th} input-output pair of the

3. Note that as ℓ_ρ-norm ($\rho > 1$) is differentiable, it rarely induce sparsity (Szafranski et al., 2010). However, as $\rho \to 1$, they promote only a few leading terms due to the high curvatures of such norms (Szafranski et al., 2007). In order to obtain a sparse solution in such cases, thresholding is commonly employed by existing ℓ_p-MKL ($\rho > 1$) algorithms (Vishwanathan et al., 2010; Orabona et al., 2012; Jain et al., 2012; Jawanpuria et al., 2014). We employed thresholding in our experiments.

t^{th} task. For the sake of notational simplicity, it is assumed that the number of training examples is same for all the tasks. The prediction function for the t^{th} task is given by: $F_t(\mathbf{x}) = \sum_{v\in\mathcal{V}} \langle f_{tv}, \phi_{k_v}(\mathbf{x})\rangle_{\mathcal{H}_{k_v}} - b_t$, where $f_t = (f_{tv})_{v\in\mathcal{V}}$ and b_t are the task parameters to be learnt. We propose the following regularized risk minimization problem for estimating these task parameters and term it as gHKL$_{\text{MT}}$:

$$\min_{f_t, b_t \forall t} \frac{1}{2} \Bigg(\underbrace{\sum_{v\in\mathcal{V}} d_v \left(\sum_{w\in D(v)} (Q_w(f_1,\ldots,f_T))^{\rho} \right)^{\frac{1}{\rho}}}_{\Omega_T(f_1,\ldots,f_T)} \Bigg)^2 + C \sum_{t=1}^{T} \sum_{i=1}^{m} \ell(y_{ti}, F_t(\mathbf{x}_{ti})), \tag{6}$$

where $\rho \in (1, 2]$ and $Q_w(f_1, \ldots, f_T)$ is a norm-based multi-task regularizer on the task parameters f_{tw} $\forall t$. In the following, we discuss the effect of the above regularization. Firstly, there is a ℓ_1-norm regularization over the group of nodes (feature spaces) and a ℓ_ρ-norm regularization within each group. This ℓ_1/ℓ_ρ block-norm regularization is same as that of gHKL and will have the same effect on the sparsity pattern of the selected feature spaces (kernels). Hence, only a few nodes (feature spaces) will be selected by the gHKL$_{\text{MT}}$ regularizer $\Omega_T(f_1, \ldots, f_T)$. Secondly, nature of the task relatedness within each (selected) feature space is governed by the $Q_w(f_1, \ldots, f_T)$ regularizer.

For instance, consider the following definition of $Q_w(f_1, \ldots, f_T)$ (Lounici et al., 2009; Jawanpuria and Nath, 2011):

$$Q_w(f_1, \ldots, f_T) = \left(\sum_{t=1}^{T} \|f_{tw}\|^2 \right)^{\frac{1}{2}}. \tag{7}$$

The above regularizer couples the task parameters within each feature space via ℓ_2-norm. It encourages the task parameters within a feature space to be either zero or non-zero across all the tasks. Therefore, $\Omega_T(f_1, \ldots, f_T)$ based on (7) has the following effect: i) all the tasks will simultaneously select or reject a given feature space, and ii) overall only a few feature spaces will be selected in the gHKL style sparsity pattern.

Several multi-task regularizations (Evgeniou and Pontil, 2004; Evgeniou et al., 2005; Jacob et al., 2008) have been proposed to encourage proximity among the task parameters within a given feature space. This correlation among the tasks may be enforced while learning a shared sparse feature space by employing the following $Q_w(f_1, \ldots, f_T)$:

$$Q_w(f_1, \ldots, f_T) = \left(\mu \left\| \frac{1}{T+\mu} \sum_{t=1}^{T} f_{tw} \right\|^2 + \sum_{t=1}^{T} \left\| f_{tw} - \frac{1}{T+\mu} \sum_{t=1}^{T} f_{tw} \right\|^2 \right)^{\frac{1}{2}}, \tag{8}$$

where $\mu > 0$ is a given parameter. The above $Q_w(f_1, \ldots, f_T)$ consists of two terms: the first regularizes the mean while the second regularizes the variance of the task parameters in the feature space induced by kernel k_w. The parameter μ controls the degree of proximity among the task parameters, with lower μ encouraging higher proximity. Note that when $\mu = \infty$, (8) simplifies to (7). The gHKL$_{\text{MT}}$ regularizer $\Omega_T(f_1, \ldots, f_T)$ based on (8) has the

following effect: i) all the tasks will simultaneously select or reject a given feature space, ii) overall only a few feature spaces will be selected in the gHKL style sparsity pattern, and iii) within each selected feature space, the task parameters $f_{tw}\ \forall t$ are in proximity.

Thus, gHKL_{MT} framework provides a mechanism to learn a shared feature space across the tasks. In addition, it can also preserve proximity among the tasks parameters in the learnt feature space. As we shall discuss in the next section, more generic correlations among task parameters may be also modeled within the gHKL_{MT} framework.

It is clear that the gHKL optimization problem (5) may be viewed as a special case of the gHKL_{MT} optimization problem (7), with the number of tasks set to unity. Hence the rest of the discussion regarding dual derivation and optimization focuses primarily on gHKL_{MT} formulation.

3.3 gHKL_{MT} Dual Formulation

As mentioned earlier, due to the presence of the ℓ_ρ-norm term in gHKL_{MT} formulation, naive extensions of the projected gradient based active set method in Bach (2008) will be rendered computationally infeasible on real world data sets. Hence, we first re-write gHKL_{MT} formulation in an elegant form, which can then be solved efficiently. To this end, we note the following variational characterization of $\Omega_T(f_1, \ldots, f_T)$.

Lemma 1 *Given $\Omega_T(f_1, \ldots, f_T)$ and $Q_w(f_1, \ldots, f_T)$ as defined in (6) and (8) respectively, we have:*

$$\Omega_T(f_1, \ldots, f_T)^2 = \min_{\gamma \in \Delta} \min_{\lambda_v \in \Delta_{\hat{\rho}}^v\ \forall v \in \mathcal{V}} \sum_{w \in \mathcal{V}} \delta_w(\gamma, \lambda)^{-1} Q_w(f_1, \ldots, f_T)^2, \tag{9}$$

where $\hat{\rho} = \frac{\rho}{2-\rho}$, $\delta_w(\gamma, \lambda)^{-1} = \sum_{v \in A(w)} \frac{d_v^2}{\gamma_v \lambda_{vw}}$, $\Delta_1 = \left\{\mathbf{z} \in \mathbb{R}^{|\mathcal{V}|} \mid \mathbf{z} \geq 0, \sum_{v \in \mathcal{V}} \mathbf{z}_v \leq 1\right\}$ and $\Delta_r^v = \left\{\mathbf{z} \in \mathbb{R}^{|D(v)|} \mid \mathbf{z} \geq 0, \sum_{w \in D(v)} \mathbf{z}_w^r \leq 1\right\}$.

Note that $\rho \in (1, 2) \Rightarrow \hat{\rho} \in (1, \infty)$. The proof of the above lemma is provided in Appendix A.2.

In order to keep the notations simple, in the remainder of this section, it is assumed that the learning tasks at hand are binary classification, i.e., $y_{ti} \in \{-1, 1\}\ \forall t, i$, and the loss function is the hinge loss. However, one can easily extend these ideas to other loss functions and learning problems. Refer Appendix A.8 for gHKL_{MT} dual formulation with general convex loss functions.

Lemma 2 *Consider problem (6) with the regularizer term replaced with its variational characterization (9) and the loss function as the hinge loss $\ell(y, F_t(\mathbf{x})) = \max(0, 1 - yF_t(\mathbf{x}))$. Then the following is a partial dual of it with respect to the variables $f_t, b_t\ \forall t = 1, \ldots, T$:*

$$\min_{\gamma \in \Delta_1} \min_{\lambda_v \in \Delta_{\hat{\rho}}^v\ \forall v \in \mathcal{V}} \max_{\alpha_t \in S(\mathbf{y}_t, C) \forall t} G(\gamma, \lambda, \alpha), \tag{10}$$

where

$$G(\gamma, \lambda, \alpha) = \mathbf{1}^\top \alpha - \frac{1}{2} \alpha^\top \mathbf{Y} \left(\sum_{w \in \mathcal{V}} \delta_w(\gamma, \lambda) H_w \right) \mathbf{Y} \alpha,$$

$\alpha = [\alpha_1^\top, \ldots, \alpha_T^\top]^\top$, $S(\mathbf{y}_t, C) = \{\beta \in \mathbb{R}^m \mid 0 \leq \beta \leq C,\ \sum_{i=1}^m y_{ti}\beta_i = 0\}$, $\mathbf{y}_t = [y_{t1}, \ldots, y_{tm}]^\top$, $\mathbf{Y}$ *is the diagonal matrix corresponding to the vector* $[\mathbf{y}_1^\top, \ldots, \mathbf{y}_T^\top]^\top$, $\mathbf{1}$ *is a* $mT \times 1$ *vector with entries as unity,* $\delta_w(\gamma, \lambda)^{-1} = \sum_{v \in A(w)} \frac{d_v^2}{\gamma_v \lambda_{vw}}$, $\Delta_1 = \left\{\mathbf{z} \in \mathbb{R}^{|\mathcal{V}|} \mid \mathbf{z} \geq 0, \sum_{v \in \mathcal{V}} \mathbf{z}_v \leq 1\right\}$, $\Delta_r^v = \left\{\mathbf{z} \in \mathbb{R}^{|D(v)|} \mid \mathbf{z} \geq 0, \sum_{w \in D(v)} \mathbf{z}_w^r \leq 1\right\}$, $\hat{\rho} = \frac{\rho}{2-\rho}$, *and* $H_w \in \mathbb{R}^{mT \times mT}$ *is the multi-task kernel matrix corresponding to the multi-task kernel* $h_w\ \forall w \in \mathcal{V}$. *The kernel function* h_w *is defined as follows:*

$$h_w(\mathbf{x}_{t_1 i}, \mathbf{x}_{t_2 j}) = k_w(\mathbf{x}_{t_1 i}, \mathbf{x}_{t_2 j}) B(t_1, t_2), \tag{11}$$

where B *is a* $T \times T$ *matrix.* $B = I$ *(identity matrix) when the multi-task regularizer (7) is employed in (6). Alternatively,* $B = I + \mathbf{1}\mathbf{1}^\top/\mu$ *(here* $\mathbf{1}$ *is a* $T \times 1$ *vector with entries as unity) in the case when the regularizer (8) is employed. The prediction function for the task* t_1 *is given by*

$$F_{t_1}(\mathbf{x}_{t_1 j}) = \sum_{t_2=1}^{T} \sum_{i=1}^{m} \bar{\alpha}_{t_2 i} y_{t_2 i} \left(\sum_{w \in \mathcal{V}} \delta_w(\bar{\gamma}, \bar{\lambda}) k_w(\mathbf{x}_{t_1 i}, \mathbf{x}_{t_2 j}) B(t_1, t_2) \right),$$

where $(\bar{\gamma}, \bar{\lambda}, \bar{\alpha})$ *is an optimal solution of (10).*

Proof The proof follows from the representer theorem (Schölkopf and Smola, 2002). Also refer to Appendix A.3. ■

This lemma shows that gHKL$_{\text{MT}}$ essentially constructs the same prediction function as an SVM with the effective multi-task kernel as: $h = \sum_{w \in \mathcal{V}} \delta_w(\gamma, \lambda) h_w$. Similarly, in the case of the gHKL, the effective kernel is $k = \sum_{w \in \mathcal{V}} \delta_w(\gamma, \lambda) k_w$ (since the terms T and B are unity). Here, as well as in the rest of the paper, we employ the symbols 'h' and 'H' for the multi-task kernel and the corresponding Gram matrix respectively.

The multi-task kernel (11) consists of two terms: the first term corresponds to the similarity between two instances $\mathbf{x}_{t_1 i}$ and $\mathbf{x}_{t_2 j}$ in the feature space induced by the kernel k_w. The second term corresponds to the correlation between the tasks t_1 and t_2. In the case of the regularizer (7), the matrix B simplifies to: $B(t_1, t_2) = 1$ if $t_1 = t_2$ and $B(t_1, t_2) = 0$ if $t_1 \neq t_2$, thereby making the kernel matrices $H_w (w \in \mathcal{V})$ block diagonal. Hence, the gHKL$_{\text{MT}}$ regularizer based on (7) promotes simultaneous sparsity in kernel selection among the tasks, without enforcing any additional correlations among the tasks.

In general, any $T \times T$ positive semi-definite matrix may be employed as B to model generic correlations among tasks. The multi-task kernel given by (11) will still remain a valid kernel (Sheldon, 2008; Álvarez et al., 2012). The matrix B is sometimes referred to as the output kernel in the setting of learning vector-valued functions. It is usually constructed from the prior domain knowledge.

We now discuss the nature of the optimal solution of (10). Most of the kernel weights $\delta_w(\gamma, \lambda)$ are zero at optimality of (10): $\delta_w(\gamma, \lambda) = 0$ whenever $\gamma_v = 0$ or $\lambda_{vw} = 0$ for any $v \in A(w)$. The vector γ is sparse due to ℓ_1-norm constraint in (10). In addition, $\rho \to 1 \Rightarrow \hat{\rho} \to 1$. Hence the vectors $\lambda_v\ \forall v \in \mathcal{V}$ get close to becoming sparse as $\rho \to 1$ due to the $\ell_{\hat{\rho}}$-norm constraint in (10). The superimposition of these two phenomena leads to a

flexible[4] sparsity pattern in kernel selection. This is explained in detail towards the end of this section.

Note that $\rho = 2 \Rightarrow \lambda_{vw} = 1 \ \forall v \in A(w), w \in \mathcal{W}$ at optimality in (10). Hence for $\rho = 2$, the minimization problem in (10) can be efficiently solved using a projected gradient method (Rakotomamonjy et al., 2008; Bach, 2009). However, as established in Liu and Ye (2010), projection onto the kind of feasibility set in the minimization problem in (10) is computationally challenging for $\rho \in (1, 2)$. Hence, we wish to re-write this problem in a relatively simpler form that can be solved efficiently. To this end, we present the following important theorem.

Theorem 3 *The following is a dual of (6) considered with the hinge loss function, and the objectives of (6) (with the hinge loss), (10) and (12) are equal at optimality:*

$$\min_{\eta \in \Delta_1} \ g(\eta), \tag{12}$$

where $g(\eta)$ is the optimal objective value of the following convex problem:

$$\max_{\alpha_t \in S(\mathbf{y}_t, C) \forall t} \ \mathbf{1}^\top \alpha - \frac{1}{2} \left(\sum_{w \in \mathcal{V}} \zeta_w(\eta) \left(\alpha^\top \mathbf{Y} H_w \mathbf{Y} \alpha \right)^{\bar{\rho}} \right)^{\frac{1}{\bar{\rho}}}, \tag{13}$$

where $\zeta_w(\eta) = \left(\sum_{v \in A(w)} d_v^\rho \eta_v^{1-\rho} \right)^{\frac{1}{1-\rho}}$, $\alpha = [\alpha_1^\top, \ldots, \alpha_T^\top]^\top$, $S(\mathbf{y}_t, C) = \{\beta \in \mathbb{R}^m \mid 0 \leq \beta \leq C, \ \sum_{i=1}^m y_{ti} \beta_i = 0\}$, $\mathbf{y}_t = [y_{t1}, \ldots, y_{tm}]^\top$, $\mathbf{Y}$ is the diagonal matrix corresponding to the vector $[\mathbf{y}_1^\top, \ldots, \mathbf{y}_T^\top]^\top$, $\mathbf{1}$ is a $mT \times 1$ vector with entries as unity, $\bar{\rho} = \frac{\hat{\rho}}{\hat{\rho}-1}$, $\hat{\rho} = \frac{\rho}{2-\rho}$, $\Delta_1 = \left\{ \mathbf{z} \in \mathbb{R}^{|\mathcal{V}|} \mid \mathbf{z} \geq 0, \sum_{v \in \mathcal{V}} \mathbf{z}_v \leq 1 \right\}$, and $H_w \in \mathbb{R}^{mT \times mT}$ is the multi-task kernel matrix corresponding to the multi-task kernel (11).

The key idea in the proof of the above theorem is to eliminate the λ variables and the details are presented in Appendix A.5. The expression for the prediction function F, in terms of the variables η and α, is provided in Appendix A.9.

This theorem provides some key insights: firstly, we have that (12) is essentially a ℓ_1-norm regularized problem and hence it is expected that most η will be zero at optimality. Since $(\eta_v = 0) \Rightarrow (\zeta_w(\eta) = 0 \ \forall w \in D(v))$, it follows that most nodes in $\mathcal{V}$ will not contribute in the optimization problems (12) and (13). Secondly, in a single task learning setting ($T = 1$), the problem in (13) is equivalent to the $\ell_{\hat{\rho}}$-norm MKL dual problem (Kloft et al., 2011) with the base kernels as $(\zeta_v(\eta))^{\frac{1}{\bar{\rho}}} k_v \ \forall v \in \mathcal{V} \ni \zeta_v(\eta) \neq 0$. The optimization problem (13) essentially learns an effective kernel of the form $h = \sum_{v \in \mathcal{V}} \theta_v \left(\zeta_v(\eta)\right)^{\frac{1}{\bar{\rho}}} h_v$, where the θ are intermediate optimization variables constrained to be non-negative and lie within a $\ell_{\hat{\rho}}$-norm ball. The expression for θ in terms of the variables η and α is provided in Appendix A.9.

The variable θ influence the nature of the effective kernel h in two important ways: i) it follows from the expression of θ that

$$\theta_v \left(\zeta_v(\eta)\right)^{\frac{1}{\bar{\rho}}} \propto \zeta_v(\eta) \left(\alpha^\top \mathbf{Y} H_v \mathbf{Y} \alpha \right)^{\frac{1}{(\hat{\rho}-1)}}.$$

4. The HKL dual formulation (Bach, 2009) is a special case of (10) with $\rho = 2$, $T = 1$ and $B = 1$. When $\rho = 2$, $\hat{\rho} = \infty$. This implies $\lambda_{vw} = 1 \ \forall \ v \in A(w), \ w \in \mathcal{V}$ at optimality, resulting in the weight bias towards kernels embedded in the ancestor nodes and restricted sparsity pattern in kernel selection

Algorithm 1 Active Set Algorithm - Outline

Input: Training data $\mathcal{D}$, the kernels (k_v) embedded on the DAG $(\mathcal{V})$, the $T \times T$ matrix B that models task correlations and tolerance ϵ.
Initialize the active set $\mathcal{W}$ with $sources(\mathcal{V})$.
Compute η, α by solving (14)
while Optimal solution for (12) is NOT obtained **do**
  Add *some* nodes to $\mathcal{W}$
  Recompute η, α by solving (14)
end while
Output: $\mathcal{W}, \eta, \alpha$

The above relation implies that the weight of the kernel h_v in the DAG $\mathcal{V}$ is not only dependent on the position[5] of the node v, but also on the suitability of the kernel h_v to the problem at hand. This helps in mitigating the kernel weight bias in favour of the nodes towards the top of the DAG from gHKL_{MT}, but which is present in HKL, and ii) as $\rho \to 1$ (and hence as $\hat{\rho} \to 1$), the optimal θ get close to becoming sparse (Szafranski et al., 2007; Orabona et al., 2012). This superimposed with the sparsity of η promotes a more flexible sparsity pattern in kernel selection that HKL, especially when $\rho \to 1$.

Next, we propose to solve the problem (12) by exploiting the sparsity pattern of the η variables and the corresponding $\zeta(\eta)$ terms at optimality. We discuss it in detail in the following section.

4. Optimization Algorithm

Note that problem (12) remains the same whether solved with the original set of variables (η) or when solved with only those $\eta_v \neq 0$ at optimality (refer Appendix A.4 for details). However the computational effort required in the latter case can be significantly lower since it involves low number of variables and kernels. This motivates us to explore an active set algorithm, which is similar in spirit to that in Bach (2008).

An outline of the proposed active set algorithm is presented in Algorithm 1. The algorithm starts with an initial guess for the set $\mathcal{W}$ such that $\eta_w \neq 0$ $(\forall w \in \mathcal{W})$ at the optimality of (12). This set $\mathcal{W}$ is called the active set. Since the weight associated with the kernel h_w will be zero whenever $\eta_v = 0$ for any $v \in A(w)$, the active set $\mathcal{W}$ must contain $sources(\mathcal{V})$, else the problem has a trivial solution. Hence, the active set is initialized with $sources(\mathcal{V})$. At each iteration of the algorithm, (12) is solved with variables restricted to those in $\mathcal{W}$:

$$\min_{\eta \in \Delta_1} \max_{\alpha_t \in S(\mathbf{y}_t, C) \forall t} \mathbf{1}^\top \alpha - \frac{1}{2} \left(\sum_{w \in \mathcal{W}} \zeta_w(\eta) \left(\alpha^\top \mathbf{Y} H_w \mathbf{Y} \alpha \right)^{\bar{\rho}} \right)^{\frac{1}{\bar{\rho}}}. \tag{14}$$

In order to formalize the active set algorithm, we need: i) an efficient algorithm for solving problem (14), ii) a condition for verifying whether a candidate solution is optimal

5. Similar to the δ_v function in HKL (3), it follows from the definition of ζ_v that $\zeta_v(\eta) \geq \zeta_w(\eta)$ $\forall w \in D(v)$ (strict inequality holds if $\zeta_w(\eta) > 0$).

Algorithm 2 Mirror Descent Algorithm for solving (14)

Input: Gram matrices H_w $(w \in \mathcal{W})$ and the regularization parameter C
Initialize $\eta_{\mathcal{W}}$ $(w \in \mathcal{W})$ such that $\eta_{\mathcal{W}} \in \Delta_1$ (warm-start may be used)
Iteration number: $i = 0$
while convergence criterion is not met[6] **do**
  $i = i + 1$
  Compute $\zeta_w(\eta_{\mathcal{W}})$ $\forall w \in \mathcal{W}$ (Theorem 3)
  Compute $\alpha_{\mathcal{W}}$ (13) using $\ell_{\hat{\rho}}$-norm MKL algorithm with kernels as $\left((\zeta_w(\eta_{\mathcal{W}}))^{\frac{1}{\hat{\rho}}} H_w\right)_{w \in \mathcal{W}}$
  Compute $\nabla g(\eta_{\mathcal{W}})$ as in (24)
  Compute step size $s = \sqrt{\log(|\mathcal{W}|)/i \cdot \|\nabla g(\eta_{\mathcal{W}}))\|_\infty^2}$
  Compute $\eta_w = \exp\left(1 + \log(\eta_w) - s \cdot \nabla g(\eta_{\mathcal{W}})_w\right)$ $\forall w \in \mathcal{W}$
  Normalize $\eta_w = \frac{\eta_w}{\sum_{v \in \mathcal{W}} \eta_v}$ $\forall w \in \mathcal{W}$
end while
Output: $\eta_{\mathcal{W}}, \alpha_{\mathcal{W}}$

with respect to the optimization problem (12), and iii) a procedure for building/improving the active set after each iteration.

We begin with the first. We propose to solve the optimization problem (14) using the mirror descent algorithm (Ben-Tal and Nemirovski, 2001; Beck and Teboulle, 2003). Mirror descent algorithm is known to efficiently solve convex programs with Lipschitz continuous and differentiable objectives constrained over a convex compact set. It achieves a near-optimal convergence rate whenever the feasibility set is a simplex (which is true in our optimization problem (14)). Mirror descent is close in spirit to the projected gradient descent algorithm and hence assumes that an oracle for computing the gradient of the objective is available.

Following the common practice of smoothing (Bach, 2009), in the rest of the paper, we employ $\zeta_w((1-\varepsilon)\eta + \frac{\varepsilon}{|\mathcal{V}|})$ instead[7] of $\zeta_w(\eta)$ in (13) with $\varepsilon > 0$. The following theorem establishes the applicability of mirror descent for solving (14):

Theorem 4 *The function $g(\eta)$ given by (13) is convex. Also, the expression for the i^{th} entry in the gradient $(\nabla g(\eta))_i$ is given in (24). If all the eigenvalues of the Gram matrices H_w are finite and non-zero, then g is Lipschitz continuous.*

The proof of the above theorem is technical and is provided in Appendix A.6.

Algorithm 2 summarizes the proposed mirror descent based algorithm for solving (14). One of its steps involve computing $\nabla g(\eta_{\mathcal{W}})$ (expression provided in (24)), which in turn requires solving (13). As noted before, (13) is similar to the $\ell_{\hat{\rho}}$-norm MKL problem (Kloft et al., 2011) but with a different feasibility set for the optimization variables α. Hence, (13) can be solved by employing a modified cutting planes algorithm (Kloft et al., 2011) or a modified sequential minimal optimization (SMO) algorithm (Platt, 1999; Vishwanathan

6. Relative objective gap between two successive iteration being less than a given tolerance ϵ is taken to be the convergence criterion. Objective here is the value of $g(\eta_{\mathcal{W}})$, calculated after $\ell_{\hat{\rho}}$-norm MKL step.
7. Note that this is equivalent to smoothing the regularizer Ω_T while preserving its sparsity inducing properties (Bach, 2009).

Algorithm 3 Active Set Algorithm

Input: Training data $\mathcal{D}$, the kernels (k_v) embedded on the DAG $(\mathcal{V})$, the $T \times T$ matrix B that models task correlations and tolerance ϵ.
Initialize the active set $\mathcal{W}$ with $sources(\mathcal{V})$
Compute η, α by solving (14) using Algorithm 2
while sufficient condition for optimality (15) is not met **do**
  Add those nodes to $\mathcal{W}$ that violate (15)
  Recompute η, α by solving (14) using Algorithm 2
end while
Output: $\mathcal{W}, \eta, \alpha$

et al., 2010). Empirically, we observed the SMO based algorithm to be much faster than the cutting planes algorithm for gHKL$_{\text{MT}}$ (and gHKL) with SVM loss functions. In the special case of $\rho = 2, T = 1$ and $B = 1$, (13) is simply a regular SVM problem.

Now we turn our attention to the second requirement of the active set algorithm: a condition to verify the optimality of a candidate solution. We present the following theorem that provides a sufficient condition for verifying optimality of a candidate solution.

Theorem 5 *Suppose the active set $\mathcal{W}$ is such that $\mathcal{W} = hull(\mathcal{W})$. Let $(\eta_{\mathcal{W}}, \alpha_{\mathcal{W}})$ be a $\epsilon_{\mathcal{W}}$-approximate optimal solution of (14), obtained from Algorithm (2). Then, it is an optimal solution for (12) with a duality gap less than ϵ if the following condition holds:*

$$\max_{u \in sources(\mathcal{W}^c)} \alpha_{\mathcal{W}}^\top \mathbf{Y} \mathcal{K}_u \mathbf{Y} \alpha_{\mathcal{W}} \leq \left(\sum_{w \in \mathcal{W}} \zeta_w(\eta_{\mathcal{W}}) \left(\alpha_{\mathcal{W}}^\top \mathbf{Y} H_w \mathbf{Y} \alpha_{\mathcal{W}} \right)^{\bar{\rho}} \right)^{\frac{1}{\bar{\rho}}} + 2(\epsilon - \epsilon_{\mathcal{W}}), \qquad (15)$$

where $\mathcal{K}_u = \sum_{w \in D(u)} \frac{H_w}{\left(\sum_{v \in A(w) \cap D(u)} d_v\right)^2}$.

The proof is provided in Appendix A.7. It closely follows that for the case of HKL (Bach, 2008).

The summary of the proposed mirror descent based active set algorithm is presented in Algorithm 3. At each iteration, Algorithm (3) verifies optimality of the current iterate by verifying the condition in (15). In case the current iterate does not satisfy this condition, the nodes in $sources(\mathcal{W}^c)$ that violate the condition (15) are included in the active set.[8] This takes care of the third requirement of the active set algorithm. The algorithm terminates if the condition (15) is satisfied by the iterate.

In the following, an estimate of the computational complexity of the active set algorithm is presented. Let W be the final active set size. The optimization problem (14) needs to be solved at most W times, assuming the worst case scenario of adding one node per active set iteration. Each run of the mirror descent algorithm requires at most $O(\log(W))$ iterations (Ben-Tal and Nemirovski, 2001; Beck and Teboulle, 2003). A conservative time complexity estimate for computing the gradient $\nabla g(\eta_{\mathcal{W}})$ by solving a variant of the $\ell_{\hat{\rho}}$-norm MKL problem (13) is $O(m^3 T^3 W^2)$. This amounts to $O(m^3 T^3 W^3 \log(W))$. As for the computational cost of the sufficient condition, let z denote the maximum out-degree

8. It is easy to see that with this update scheme, $\mathcal{W}$ is always equal to $hull(\mathcal{W})$, as required in Theorem 5.

of a node in $\mathcal{G}$, i.e., z is an upper-bound on the the maximum number of children of any node in $\mathcal{G}$. Then the size of $sources(\mathcal{W}^c)$ is upper-bounded by Wz. Hence, a total of $O(\omega m^2 T^2 W z)$ operations are required for evaluating the matrices $\mathcal{K}$ in (15), where ω is the complexity of computing a single entry in any $\mathcal{K}$. In all the pragmatic examples of kernels and the corresponding DAGs provided by Bach (2008), ω is polynomial in the training set dimensions. Moreover, caching of $\mathcal{K}$ usually renders ω to be a constant (Bach, 2009). Further, the total cost of the quadratic computation in (15) is $O(m^2 T^2 W^2 z)$. Thus the overall computational complexity is $O(m^3 T^3 W^3 \log(W) + \omega m^2 T^2 W z + m^2 T^2 W^2 z)$. More importantly, because the sufficient condition for optimality (Theorem 5) is independent of ρ, we have the following result:

Corollary 6 *In a given input setting, HKL algorithm converges in time polynomial in the size of the active set and the training set dimensions if and only if the proposed mirror descent based active set algorithm (i.e., gHKL$_{\mathrm{MT}}$ algorithm) has a polynomial time convergence in terms of the active set and training set sizes.*

The proof is provided in Appendix A.10.

In the next section, we present an application of the proposed formulation that illustrate the benefits of the proposed generalizations over HKL.

5. Rule Ensemble Learning

In this section, we propose a solution to the problem of learning an ensemble of decision rules, formally known as Rule Ensemble Learning (REL) (Cohen and Singer, 1999), employing the gHKL and gHKL$_{\mathrm{MT}}$ formulations. For the sake of simplicity, we only discuss the single task REL setting in this section, i.e., REL as an application of gHKL. Similar ideas can be applied to perform REL in multi-task learning setting, by employing gHKL$_{\mathrm{MT}}$. In fact, we present empirical results of REL in both single and multiple task learning settings in Section 6. We begin with a brief introduction to REL.

If-then decision rules (Rivest, 1987) are one of the most expressive and human readable representations for learned hypotheses. It is a simple logical pattern of the form: IF *condition* THEN *decision*. The *condition* consists of a conjunction of a small number of simple boolean statements (propositions) concerning the values of the individual input variables while the *decision* specifies a value of the function being learned. An instance of a decision rule from Quinlan's play-tennis example (Quinlan, 1986) is:

IF *HUMIDITY==normal* AND *WIND==weak* THEN *PlayTennis==yes*.

The dominant paradigm for induction of rule sets, in the form of decision list (DL) models for classification (Rivest, 1987; Michalski, 1983; Clark and Niblett, 1989), has been a greedy *sequential covering* procedure.

REL is a general approach that treats decision rules as base classifiers in an ensemble. This is in contrast to the more restrictive decision list models that are disjunctive sets of rules and use only one in the set for each prediction. As pointed out in Cohen and Singer (1999), boosted rule ensembles are in fact simpler, better-understood formally than other state-of-the-art rule learners and also produce comparable predictive accuracy.

REL approaches like SLIPPER (Cohen and Singer, 1999), LRI (Weiss and Indurkhya, 2000), RuleFit (Friedman and Popescu, 2008), ENDER/MLRules (Dembczyński et al., 2008, 2010) have additionally addressed the problem of learning a compact set of rules that generalize well in order to maintain their readability. Further, a number of rule learners like RuleFit, LRI encourage shorter rules (i.e., fewer conjunctions in the condition part of the rule) or rules with a restricted number of conjunctions, again for purposes of interpretability. We build upon this and define our REL problem as that of learning a small set of simple rules and their weights that leads to a good generalization over new and unseen data. The next section introduces the notations and the setup in context of REL.

5.1 Notations and Setup

Let $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$ be the training data described using p basic (boolean) propositions, i.e., $\mathbf{x}_i \in \{0,1\}^p$. In case the input features are not boolean, such propositions can be derived using logical operators such as $==, \neq, \leq$ or $\geq$ over the input features (refer Friedman and Popescu, 2008; Dembczyński et al., 2008, for details). Let $\mathcal{V}$ be an index-set for all possible conjunctions with the p basic propositions and let $\phi_v : \mathbb{R}^n \mapsto \{0,1\}$ denote the v^{th} conjunction in $\mathcal{V}$. Let $f_v \in \mathbb{R}$ denote the weight for the conjunction ϕ_v. Then, the rule ensemble to be learnt is the weighted combination of these conjunctive rules: $F(\mathbf{x}) = \sum_{v \in \mathcal{V}} f_v \phi_v(\mathbf{x}) - b$, where perhaps many weights (f_v) are equal to zero.

One way to learn the weights is by performing a ℓ_1-norm regularized risk minimization in order to select few promising conjunctive rules (Friedman and Popescu, 2008; Dembczyński et al., 2008, 2010). However, to the best of our knowledge, rule ensemble learners that identify the need for sparse f, either approximate such a regularized solution using strategies such as shrinkage (Rulefit, ENDER/MLRules) or resort to post-pruning (SLIPPER). This is because the size of the minimization problem is exponential in the number of basic propositions and hence the problem becomes computationally intractable with even moderately sized data sets. Secondly, conjunctive rules involving large number of propositions might be selected. However, such conjunctions adversely effect the interpretability. We present an approach based on the gHKL framework that addresses these issues.

We begin by noting that $\langle \mathcal{V}, \subseteq \rangle$ is a subset-lattice; hereafter this will be referred to as the *conjunction lattice*. In a conjunction lattice, $\forall\ v_1, v_2 \in \mathcal{V},\ v_1 \subseteq v_2$ *if and only if* the set of propositions in conjunction v_1 is a subset of those in conjunction v_2. As an example, (*HUMIDITY==normal*) is considered to be a subset of (*HUMIDITY==normal* AND *WIND==weak*). The top node of this lattice is a node with no conjunctions and is also *sources*($\mathcal{V}$). Its children, the second level nodes, are all the basic propositions, p in number. The third level nodes, children of these basic propositions, are the conjunctions of length two and so on. The bottom node at $(p+1)^{th}$ level is the conjunction of all basic propositions. The number of different conjunctions of length r is $\binom{p}{r}$ and the total number of nodes in this *conjunction lattice* is 2^p. Figure (1) shows a complete conjunction lattice with $p = 4$.

We now discuss how the proposed gHKL regularizer (5) provides an efficient and optimal solution to a regularized empirical risk minimization formulation for REL.

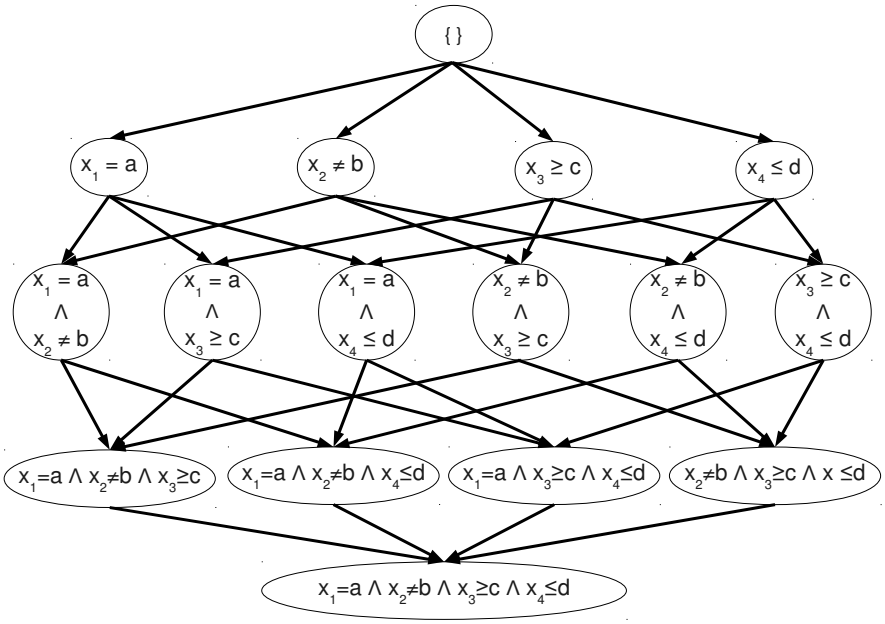


Figure 1: Example of a conjunction lattice with 4 basic propositions: $(x_1 = a)$, $(x_2 \neq b)$, $(x_3 \geq c)$ and $(x_4 \leq d)$. The input space consist of four features: x_1, x_2, x_3 and x_4. The number of nodes in conjunction lattice is exponential in the number of basic propositions. In this particular example, the number of nodes is 16 $(= 2^4)$.

5.2 Rule Ensemble Learning with gHKL

The key idea is to employ gHKL formulation (5) with the DAG as the conjunction lattice and the kernels as $k_v(\mathbf{x}_i, \mathbf{x}_j) = \phi_v(\mathbf{x}_i)\phi_v(\mathbf{x}_j)$ for learning an ensemble of rules. Note that with such a setup, the ℓ_1/ℓ_ρ block-norm regularizer in gHKL ($\Omega_S(f) = \sum_{v\in\mathcal{V}} d_v \|f_{D(v)}\|_\rho$) implies: 1) for most $v \in \mathcal{V}$, $f_v = 0$, and 2) for most $v \in \mathcal{V}$, $f_w = 0 \; \forall \; w \in D(v)$. In the context of the REL problem, the former statement is equivalent to saying: selection of a compact set of conjunctions is promoted, while the second reads as: selection of conjunctive rules with small number of propositions is encouraged. Thus, gHKL formulation constructs a compact ensemble of simple conjunctive rules. In addition, we set $d_v = a^{|S_v|}$ $(a > 1)$, where S_v is the set of basic propositions involved in the conjunction ϕ_v. Such a choice further encourages selection of short conjunctions and leads to the following elegant computational result:

Theorem 7 *The complexity of the proposed gHKL algorithm in solving the REL problem, with the DAG, the base kernels and the parameters d_v as defined above, is polynomial in the size of the active set and the training set dimensions. In particular, if the final active set size is W, then its complexity is given by $O(m^3W^3\log(W) + m^2W^2p)$.*

The proof is provided in Appendix A.11.

We end this section by noting the advantage of the generic regularizer in gHKL formulation over the that in HKL formulation in the context of REL application. Recall that

the sparsity pattern allowed by HKL has the following consequence: a conjunction is selected only after selecting all the conjunctions which are subsets of it. This, particularly in the context of REL, is psycho-visually redundant, because a rule with k propositional statements, if included in the result, will necessarily entail the inclusion of $(2^k - 1)$ more general rules in the result. This violates the important requirement for a small set (Friedman and Popescu, 2008; Dembczyński et al., 2008, 2010) of human-readable rules. The gHKL regularizer, with $\rho \in (1, 2)$, alleviates this restriction by promoting additional sparsity in selecting the conjunctions. We empirically evaluate the proposed gHKL based solution for REL application in the next section.

6. Experimental Results

In this section, we report the results of simulation in REL on several benchmark binary and multiclass classification data sets from the UCI repository (Blake and Lichman, 2013). The goal is to compare various rule ensemble learners on the basis of: (a) generalization, which is measured by the predictive performance on unseen test data, and (b) ability to provide compact set of simple rules to facilitate their readability and interpretability (Friedman and Popescu, 2008; Dembczyński et al., 2010; Cohen and Singer, 1999). The latter is judged using i) average number of rules learnt, and ii) average number of propositions per rule. The following REL approaches were compared.

- **RuleFit:** Rule ensemble learning algorithm proposed by Friedman and Popescu (2008). All the parameters were set to the default values mentioned by the authors. In particular, the model was set in the mixed linear-rule mode, average tree size was set 4 and maximum number of trees were kept as 500. The same configuration was also used by Dembczyński et al. (2008, 2010) in their simulations. This REL system cannot handle multi-class data sets and hence is limited to the simulations on binary classification data sets. Its code is available at `www-stat.stanford.edu/~jhf/R-RuleFit.html`.

- **SLI:** The SLIPPER algorithm proposed by Cohen and Singer (1999). Following Dembczyński et al. (2008, 2010), all parameters were set to their defaults. We retained the internal cross-validation for selecting the optimal number of rules.

- **ENDER:** State-of-the-art rule ensemble learning algorithm (Dembczyński et al., 2010). For classification setting, `ENDER` is same as `MLRules` (Dembczyński et al., 2008). The parameters were set to the default values suggested by the authors. The second order heuristic was used for minimization. Its code is available at `www.cs.put.poznan.pl/wkotlowski`.

- **HKL-ℓ_1-MKL:** A two-stage rule ensemble learning approach. In the first stage, HKL is employed to prune the exponentially large search space of all possible conjunctive rules and select a set of candidate rules (kernels). The rule ensemble is learnt by employing ℓ_1-MKL over the candidate set of rules. In both the stages, a three-fold cross validation procedure was employed to tune the C parameter with values in $\{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$.

- **gHKL$_\rho$:** The proposed gHKL based REL formulation for binary classification problem. We considered three different values of ρ: 2, 1.5 and 1.1. Note that for binary

classification, $\rho = 2$ renders the HKL formulation (Bach, 2008). In each case, a three-fold cross validation procedure was employed to tune the C parameter with values in $\{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$. As mentioned earlier, the parameters $d_v = 2^{|v|}$.

- **gHKL$_{\mathbf{MT}-\rho}$:** The proposed gHKL$_{\mathrm{MT}}$ based REL formulation for multiclass classification problem. For each class, a one-vs-rest binary classification task is created. Since we did not have any prior knowledge about the correlation among the classes in the data sets, we employed the multi-task regularizer (7) in the gHKL$_{\mathrm{MT}}$ primal formulation (6).

We considered three different values of ρ: 2, 1.5 and 1.1. Its parameters and cross validation details are same as that of gHKL$_\rho$. The implementations of both gHKL$_\rho$ and gHKL$_{\mathtt{MT}-\rho}$ are available at `http://www.cse.iitb.ac.in/~pratik.j/ghkl`.

Note that the above methods differ in the way they control the number of rules (M) in the ensemble. In the case of gHKL$_\rho$ (gHKL$_{\mathtt{MT}-\rho}$), M implicitly depends on the parameters: ρ, C and d_v. SLI has a parameter for maximum number of rules M_{max} and M is decided via a internal cross-validation such that $M \leq M_{max}$. For the sake of fairness in comparison with gHKL$_\rho$, we set $M_{max} = \max(M_{1.5}, M_{1.1})$, where M_ρ is the average number of rules obtained with gHKL$_\rho$ (gHKL$_{\mathtt{MT}-\rho}$). ENDER has an explicit parameter for the number of rules, which is also set to $\max(M_{1.5}, M_{1.1})$. In case of RuleFit, the number of rules in the ensemble is determined internally and is not changed by us.

6.1 Binary Classification in REL

This section summarizes our results on binary REL classification. Table 1 provides the details of the binary classification data sets. For every data set, we created 10 random train-test splits with 10% train data (except for MONK-3 data set, whose train-test split of $122 - 432$ instances respectively was already given in the UCI repository). Since many data sets were highly unbalanced, we report the average F1-score along with the standard deviation (Table 5 in Appendix A.12 reports the average AUC). The results are presented in Table 2. The best result, in terms of the average F1-score, for each data set is highlighted.

Data set		Num	Bias	p	$\lvert\mathcal{V}\rvert$	Data set		Num	Bias	p	$\lvert\mathcal{V}\rvert$
TIC-TAC-TOE	(TIC)	958	1.89	54	$\approx 10^{16}$	HEARTSTAT	(HTS)	270	0.8	76	$\approx 10^{22}$
B-CANCER-W	(BCW)	699	0.53	72	$\approx 10^{21}$	MONK-3	(MK3)	554	1.08	30	$\approx 10^{9}$
DIABETES	(DIA)	768	0.54	64	$\approx 10^{19}$	VOTE	(VTE)	232	0.87	32	$\approx 10^{9}$
HABERMAN	(HAB)	306	0.36	40	$\approx 10^{12}$	B-CANCER	(BCC)	277	0.41	76	$\approx 10^{22}$
HEARTC	(HTC)	296	0.85	78	$\approx 10^{23}$	MAM. MASS	(MAM)	829	0.94	46	$\approx 10^{13}$
BLOOD TRANS	(BLD)	748	3.20	32	$\approx 10^{9}$	LIVER	(LIV)	345	1.38	48	$\approx 10^{14}$

Table 1: Data sets used for binary REL classification. 'Num' is the number of instances in the data set while 'Bias' denotes the ratio of # of +ve and −ve instances. The number of number of basic propositions is 'p' and $|\mathcal{V}|$ represents the total number of possible conjunctions. For each numerical input feature, 8 basic propositions were derived. The letters in brackets are the acronym used for the corresponding data set in Table 2.

	RuleFit	**SLI**	**ENDER**	**HKL-ℓ_1-MKL**	**gHKL$_\rho$**		
					$\rho = 2$	$\rho = 1.5$	$\rho = 1.1$
TIC	0.517 ± 0.092 (57.7, 2.74)	0.665 ± 0.053 (10.3, 1.96)	0.668 ± 0.032 (187, 3.17)	0.749 ± 0.040 (74.8, 1.89)	0.889 ± 0.093 (161.7, 1.72)	0.897 ± 0.093 (186.6, 1.76)	$\mathbf{0.905 \pm 0.096}^*$ (157.6, 1.72)
BCW	0.879 ± 0.025 (17.5, 2.03)	$\mathbf{0.928 \pm 0.018}$ (4.4, 1.15)	0.900 ± 0.041 (21, 1.56)	0.925 ± 0.032 (27,1.03)	0.923 ± 0.032 (30.9, 1)	0.924 ± 0.032 (20, 1.03)	0.925 ± 0.032 (20.4, 1.02)
DIA	0.428 ± 0.052 (32.9, 2.66)	0.659 ± 0.027 (4.9, 1.42)	0.656 ± 0.027 (74.0, 2.65)	0.658 ± 0.028 (47.6, 1.40)	0.661 ± 0.018 (83.2, 1.31)	$\mathbf{0.663 \pm 0.017}$ (73.2, 1.17)	0.661 ± 0.023 (62.6, 1.27)
HAB	0.175 ± 0.079 (7.5, 1)	0.483 ± 0.057 (2.1, 1)	0.474 ± 0.057 (52, 3.59)	0.506 ± 0.048 (45.6, 1.48)	$\mathbf{0.523 \pm 0.062}$ (112.1, 1.366)	0.521 ± 0.060 (51.2, 1.235)	0.521 ± 0.060 (17.1, 1.142)
HTC	0.581 ± 0.047 (8.8, 1)	0.727 ± 0.05 (3.2, 1.23)	0.724 ± 0.032 (32, 2.05)	$\mathbf{0.750 \pm 0.038}$ (32.9, 1.09)	0.743 ± 0.038 (46.7, 1.06)	0.735 ± 0.058 (23.9, 1)	0.736 ± 0.055 (32, 1.09)
BLD	0.163 ± 0.088 (40.7, 2.26)	0.476 ± 0.057 (2.0, 1)	0.433 ± 0 (63, 1.97)	0.572 ± 0.029 (175.9, 2.13)	0.586 ± 0.029 (229.7, 1.98)	0.587 ± 0.028 (62.8, 1.79)	$\mathbf{0.588 \pm 0.027}$ (19, 1.29)
HTS	0.582 ± 0.040 (9.3, 1)	0.721 ± 0.065 (3.5, 1.07)	0.713 ± 0.055 (25, 2.02)	$\mathbf{0.752 \pm 0.036}$ (24.6, 1.06)	0.747 ± 0.031 (34.7, 1.02)	0.746 ± 0.028 (25, 1.02)	0.747 ± 0.028 (24.4, 1.03)
MK3	0.947 (52, 2.88)	0.802 (1, 3)	**0.972** (93, 1.96)	**0.972** (17, 1.88)	**0.972** (200, 2.07)	**0.972** (93, 1.84)	**0.972** (7, 1.43)
VTE	0.913 ± 0.047 (2.7, 1)	0.935 ± 0.055 (1.3, 1.15)	$\mathbf{0.951 \pm 0.035}$ (9, 1.07)	0.927 ± 0.045 (23.5, 1.17)	0.93 ± 0.042 (39, 1.11)	0.929 ± 0.043 (8.2, 1)	0.934 ± 0.038 (6.4, 1)
BCC	0.254 ± 0.089 (8.1, 1)	0.476 ± 0.086 (1.2, 1)	0.452 ± 0.079 (31, 2.93)	$\mathbf{0.588 \pm 0.057}$ (33.6, 1.17)	0.565 ± 0.059 (39.6, 1.15)	0.563 ± 0.061 (30.2, 1.07)	0.569 ± 0.063 (29.4, 1.17)
MAM	0.668 ± 0.032 (26.4, 2.68)	0.808 ± 0.022 (5.3, 1.43)	$\mathbf{0.816 \pm 0.018}$ (48, 2.53)	0.805 ± 0.028 (38.7, 1.32)	0.796 ± 0.026 (92.2, 1.27)	0.796 ± 0.026 (47.6, 1.24)	0.797 ± 0.024 (40.5, 1.25)
LIV	0.357 ± 0.016 (10, 1)	0.445 ± 0.083 (1.5, 1)	0.563 ± 0.058 (59, 2.35)	0.585 ± 0.071 (43.4, 1.56)	0.594 ± 0.046 (242.5, 1.42)	$\mathbf{0.595 \pm 0.048}$ (58.2, 1.32)	0.588 ± 0.049 (45.7, 1.36)

Table 2: Results on binary REL classification. We report the F1-score along with standard deviation and, in brackets below, the number of the learnt rules as well as the average length of the learnt rules. The proposed REL algorithm, gHKL$_\rho$ ($\rho = 1.5, 1.1$), obtains better generalization performance than state-of-the-art ENDER in most data sets, with the additional advantage of learning a smaller set of more compact rules. The '*' symbol denotes statistically significant improvement. The results are averaged over ten random train-test splits.

Additionally if the best result achieves a statistically significant improvement over its nearest competitor, it is marked with a '*'. Statistical significance test is performed using the paired t-test at 99% confidence. We also report the average number of rules learnt (r) and the average length of the rules (c), specified below each F1-score as: (r, c). As discussed earlier, it is desirable that REL algorithms achieve high F1-score with a compact set of simple rules, i.e., low r and c.

We can observe from Table 2 that $\texttt{gHKL}_\rho$ obtains better generalization performance than state-of-the-art ENDER in most of the data sets with the additional advantage of having rules with smaller number of conjunctions. In fact, when averaged over the data sets, $\texttt{gHKL}_{1.1}$ and $\texttt{gHKL}_{1.5}$ output the shortest rules among all the methods. $\texttt{gHKL}_{1.1}$ obtains statistically significant performance in TIC-TAC-TOE data set. Though the generalization obtained by $\texttt{gHKL}_2$ (HKL), $\texttt{gHKL}_{1.5}$ and $\texttt{gHKL}_{1.1}$ are similar, the number of rules selected by $\texttt{gHKL}_2$ is always higher than $\texttt{gHKL}_{1.1}$ (by as much as 25 times in a few cases), hampering its interpretability.

6.2 Multiclass Classification in REL

This section summarizes our results on multiclass REL classification. The details of the multiclass data sets are provided in Table 3. Within the data sets, classes with too few instances (< 3) were not considered for simulations since we perform a three-fold cross validation for hyper-parameter selection. The results, averaged over 10 random train-test splits with 10% train data are presented in Table 4. Following Dembczyński et al. (2008, 2010), we report the accuracy to compare generalization performance among the algorithms. The number of rules as well as the average length of the rules is also reported to judge the interpretability of the output.

We can observe that $\texttt{gHKL}_{\texttt{MT}-\rho}$ obtains the best generalization performance in seven data sets, out of which four are statistically significant. Moreover, $\texttt{gHKL}_{\texttt{MT}-1.5}$ and $\texttt{gHKL}_{\texttt{MT}-1.1}$ usually select the shortest rules among all the methods. The number of rules as well as the average rule length of $\texttt{gHKL}_{\texttt{MT}-2}$ is generally very large compared to $\texttt{gHKL}_{\texttt{MT}-1.5}$ and $\texttt{gHKL}_{\texttt{MT}-1.1}$. This again demonstrates the suitability of the proposed ℓ_1/ℓ_ρ regularizer in obtaining a compact set of simple rules.

Data set	Num	c	p	$\lvert\mathcal{V}\rvert$	Data set	Num	c	p	$\lvert\mathcal{V}\rvert$
BALANCE	625	3	32	$\approx 10^9$	IRIS	150	3	50	$\approx 10^{15}$
CAR	1728	4	42	$\approx 10^{12}$	LYMPH	146	3	86	$\approx 10^{25}$
C.M.C.	1473	3	54	$\approx 10^{16}$	T.A.E.	151	3	114	$\approx 10^{34}$
ECOLI	332	6	42	$\approx 10^{12}$	YEAST	1484	10	54	$\approx 10^{16}$
GLASS	214	6	72	$\approx 10^{21}$	ZOO	101	7	42	$\approx 10^{12}$

Table 3: Data sets used for multiclass REL classification. 'Num' is the number of instances in the data set while 'c' denotes the number of classes. The number of number of basic propositions is 'p' and $|\mathcal{V}|$ represents the total number of possible conjunctions. For each numerical input feature, 8 basic propositions were derived.

	SLI	**ENDER**	**gHKL$_{\mathbf{MT}-\rho}$**		
			$\rho = 2$	$\rho = 1.5$	$\rho = 1.1$
BALANCE	0.758 ± 0.025 (10.4, 1.7)	0.795 ± 0.034 (112, 2.4)	$\mathbf{0.817 \pm 0.028}$ (2468.9, 2.84)	0.808 ± 0.032 (112, 1.64)	0.807 ± 0.034 (85, 1.61)
CAR	0.823 ± 0.029 (18.3, 2.93)	0.835 ± 0.024 (270, 3.05)	0.864 ± 0.020 (9571.2, 3.14)	0.86 ± 0.028 (220.3, 1.64)	$\mathbf{0.875 \pm 0.029}^*$ (269.3, 1.85)
C.M.C.	0.446 ± 0.016 (21.1, 1.9)	$\mathbf{0.485 \pm 0.015}^*$ (513, 4.36)	0.472 ± 0.014 (10299.3, 2.85)	0.463 ± 0.017 (512.9, 1.95)	0.465 ± 0.016 (396.4, 1.88)
ECOLI	0.726 ± 0.042 (7.8, 1.34)	0.636 ± 0.028 (35, 2.15)	0.779 ± 0.057 (4790.2, 2.99)	$\mathbf{0.784 \pm 0.045}^*$ (34.3, 1.05)	0.778 ± 0.054 (32.4, 1.16)
GLASS	0.43 ± 0.061 (7.4, 1.41)	0.465 ± 0.052 (70, 3.21)	0.501 ± 0.049 (5663.7, 2.40)	$\mathbf{0.525 \pm 0.043}^*$ (69.1, 1.15)	0.524 ± 0.046 (54.6, 1.04)
IRIS	0.766 ± 0.189 (2.2, 1.02)	0.835 ± 0.093 (10, 1.34)	0.913 ± 0.083 (567, 2.44)	$\mathbf{0.927 \pm 0.024}^*$ (9.8, 1)	0.893 ± 0.091 (8.6, 1)
LYMPH	0.61 ± 0.066 (2.7, 1)	0.706 ± 0.058 (34, 2.2)	0.709 ± 0.061 (4683.8, 2.30)	$\mathbf{0.724 \pm 0.078}$ (33.7, 1.01)	0.722 ± 0.078 (33, 1.01)
T.A.E.	0.334 ± 0.035 (1.1, 1)	0.41 ± 0.065 (39, 1.86)	$\mathbf{0.418 \pm 0.049}$ (5707.4, 2.25)	0.399 ± 0.049 (38.3, 1.00)	0.402 ± 0.046 (38.1, 1.05)
YEAST	0.478 ± 0.035 (23.4, 1.63)	$\mathbf{0.497 \pm 0.015}$ (218, 5.78)	0.487 ± 0.021 (8153.6, 2.85)	0.485 ± 0.022 (217.8, 1.80)	0.486 ± 0.021 (179.6, 1.73)
ZOO	0.556 ± 0.062 (7.1, 1.24)	$\mathbf{0.938 \pm 0.033}$ (33, 1.29)	0.877 ± 0.06 (3322.2, 2.70)	0.928 ± 0.037 (32.3, 1.00)	0.927 ± 0.039 (31.9, 1.01)

Table 4: Results on multiclass REL classification. We report the accuracy along with standard deviation and, in the brackets below, the number of learnt rules as well as the average length of the learnt rules. The proposed REL algorithm, gHKL$_{\mathtt{MT}-\rho}$, obtains the best generalization performance in most data sets. In addition, for $\rho = 1.5$ and 1.1, our algorithm learns a smaller set of more compact rules than state-of-the-art ENDER. The '*' symbol denotes statistically significant improvement. The results are averaged over ten random train-test splits.

7. Summary

This paper generalizes the HKL framework in two ways. First, a generic ℓ_1/ℓ_ρ block-norm regularizer, $\rho \in (1, 2)$, is employed that provides a more flexible kernel selection pattern than HKL by mitigating the weight bias towards the kernels that are nearer to the sources of the DAG. Secondly, the framework is further generalized to the setup of learning a shared feature representation among multiple related tasks. We pose the problem of learning shared features across the tasks as that of learning a shared kernel. An efficient mirror descent based active set algorithm is proposed to solve the generalized formulations (gHKL/gHKL$_{\mathrm{MT}}$). An interesting computational result is that gHKL/gHKL$_{\mathrm{MT}}$ can be solved in time polynomial in the active set and training set sizes whenever the HKL formulation can be solved in polynomial time. The other important contribution in this paper is the application of the proposed gHKL/gHKL$_{\mathrm{MT}}$ formulations in the setting of Rule Ensemble Learning (REL),

where HKL has not been previously explored. We pose the problem of learning an ensemble of propositional rules as a kernel learning problem. Empirical results on binary as well as multiclass classification for REL demonstrate the effectiveness of the proposed generalizations.

Acknowledgments

We thank the anonymous reviewers for the valuable comments. We acknowledge Chiranjib Bhattacharyya for initiating discussions on optimal learning of rule ensembles. Pratik Jawanpuria acknowledges support from IBM Ph.D. fellowship.

Appendix A.

In the appendix section, we provide the proofs of theorems/lemmas referred to in the main paper.

A.1 Lemma 26 of Micchelli and Pontil (2005)

Let $a_i \geq 0, i = 1, \ldots, d$, $1 \leq r < \infty$ and $\Delta_{d,r} = \left\{ \mathbf{z} \in \mathbb{R}^d \mid \mathbf{z} \geq 0, \sum_{i=1}^d \mathbf{z}_i^r \leq 1 \right\}$. Then, the following result holds:

$$\min_{\mathbf{z} \in \Delta_{d,r}} \sum_{i=1}^d \frac{a_i}{\mathbf{z}_i} = \left(\sum_{i=1}^d a_i^{\frac{r}{r+1}} \right)^{1+\frac{1}{r}}.$$

The minimum is attained at

$$\mathbf{z}_i = \frac{a_i^{\frac{1}{r+1}}}{\left(\sum_{j=1}^d a_j^{\frac{r}{r+1}} \right)^{\frac{1}{r}}} \quad \forall i = 1, \ldots, d.$$

The proof follows from Holder's inequality.

A.2 Proof of Lemma 1

Proof Applying the above lemma (Appendix A.1) on the outermost ℓ_1-norm of the regularizer $\Omega_T(f_1, \ldots, f_T)^2$ in (6), we get

$$\Omega_T(f_1, \ldots, f_T)^2 = \min_{\gamma \in \Delta_1} \sum_{v \in \mathcal{V}} \frac{d_v^2}{\gamma_v} \left(\sum_{w \in D(v)} (Q_w(f_1, \ldots, f_T))^\rho \right)^{\frac{2}{\rho}},$$

where $\Delta_1 = \left\{ \mathbf{z} \in \mathbb{R}^{|\mathcal{V}|} \mid \mathbf{z} \geq 0, \sum_{v \in \mathcal{V}} \mathbf{z}_v \leq 1 \right\}$. Reapplying the above lemma on the individual terms of the above summation gives

$$\left(\sum_{w \in D(v)} (Q_w(f_1, \ldots, f_T)^2)^{\frac{\rho}{2}} \right)^{\frac{2}{\rho}} = \min_{\lambda_v \in \Delta_\rho^v} \sum_{w \in D(v)} \frac{Q_w(f_1, \ldots, f_T)^2}{\lambda_{vw}},$$

where $\hat{\rho} = \frac{\rho}{2-\rho}$ and $\Delta_r^v = \left\{ \mathbf{z} \in \mathbb{R}^{|D(v)|} \mid \mathbf{z} \geq 0, \sum_{w \in D(v)} \mathbf{z}_w^r \leq 1 \right\}$. Using the above two results and regrouping the terms will complete the proof. ■

A.3 Re-parameterization of the Multi-task Regularizer in (8)

The gHKL$_{\text{MT}}$ dual formulation (10) follows from the representer theorem (Schölkopf and Smola, 2002) after employing the following re-parameterization in (8).

Define $f^{0w} = \frac{1}{T+\mu} \sum_{t=1}^{T} f_{tw}$ and $f^{tw} = f_{tw} - f^{0w}$. Then, $Q_w(f_1, \ldots, f_T)$ in (8) may be rewritten as:

$$Q_w(f_1, \ldots, f_T) = \left(\mu \|f^{0w}\|^2 + \sum_{t=1}^{T} \|f^{tw}\|^2 \right)^{\frac{1}{2}}.$$

Further, construct the following feature map (Evgeniou and Pontil, 2004)

$$\Phi_w(\mathbf{x}, t) = \Big(\frac{\phi_w(\mathbf{x})}{\sqrt{\mu}}, \underbrace{\mathbf{0}, \ldots, \mathbf{0}}_{\text{for tasks before t}}, \phi_w(\mathbf{x}), \underbrace{\mathbf{0}, \ldots, \mathbf{0}}_{\text{for tasks after t}} \Big) \tag{16}$$

and define $f_w = (\sqrt{\mu} f^{0w}, f^{1w}, \ldots, f^{Tw})$.

With the above definitions, we rewrite the gHKL$_{\text{MT}}$ primal regularizer as well as the prediction function: $Q_w(f_1, \ldots, f_T)^2 = \|f_w\|^2$ and $F_t(\mathbf{x}) = \sum_{w \in \mathcal{V}} \langle f_w, \Phi_w(\mathbf{x}, t) \rangle - b_t \ \forall\ t$. It follows from Lemma 1 that the gHKL$_{\text{MT}}$ primal problem based on (8) is equivalent to the following optimization problem:

$$\min_{\gamma \in \Delta_1} \ \min_{\lambda_v \in \Delta_\rho^v \ \forall v \in \mathcal{V}} \ \min_{f, b} \ \frac{1}{2} \sum_{w \in \mathcal{V}} \delta_w(\gamma, \lambda)^{-1} \|f_w\|^2 + C \sum_{t=1}^{T} \sum_{i=1}^{m} \ell(y_{ti}, F_t(\mathbf{x}_{ti})), \tag{17}$$

where $f = (f_w)_{w \in \mathcal{V}}$ and $b = [b_1, \ldots, b_T]$.

A.4 Motivation for the Active Set Algorithm

Lemma 8 *The problem (12) remains the same whether solved with the original set of variables (η) or when solved with only those $\eta_v \neq 0$ at optimality.*

Proof The above follows from the following reasoning: a) variables η owe their presence in (12) only via $\zeta(\eta)$ functions, b) $(\eta_v = 0) \Rightarrow (\zeta_w(\eta) = 0 \ \forall w \in D(v))$, c) Let (η', α') be an optimal solution of the problem (12). If $\zeta_v(\eta') = 0$ and $\eta'_v \neq 0$, then (η^*, α') is also an optimal solution of the problem (12) where $\eta^*_w = \eta'_w \ \forall w \in \mathcal{V} \setminus v$ and $\eta^*_v = 0$, and d) min-max interchange in (12) yields an equivalent formulation. ■

Lemma 9 *The following min-max interchange is equivalent:*

$$\min_{\eta \in \Delta_1} \ \max_{\alpha_t \in S(\mathbf{y}_t, C) \forall t} \bar{G}(\eta, \alpha) = \max_{\alpha_t \in S(\mathbf{y}_t, C) \forall t} \ \min_{\eta \in \Delta_1} \bar{G}(\eta, \alpha),$$

where

$$\bar{G}(\eta,\alpha)=\mathbf{1}^\top\alpha-\frac{1}{2}\left(\sum_{w\in\mathcal{V}}\zeta_w(\eta)\left(\alpha^\top\mathbf{Y}H_w\mathbf{Y}\alpha\right)^{\bar{\rho}}\right)^{\frac{1}{\bar{\rho}}}.$$

Proof Note that $G(\eta,\alpha)$ is a convex function in η and a concave function in α. The min-max interchange follows from Sion-Kakutani minimax theorem (Sion, 1958). ■

A.5 Proof of Theorem 3

Before stating the proof of Theorem 3, we first prove the results in Lemma 10, Proposition 11 and Lemma 12, which will be employed therein (also see Bach, 2009, Lemma 10 and Proposition 11).

Lemma 10 *Let $a_i>0\ \forall i=1,\ldots,d$, $1<r<\infty$ and $\Delta_1=\left\{\mathbf{z}\in\mathbb{R}^d\mid\mathbf{z}\geq 0,\sum_{i=1}^d\mathbf{z}_i\leq 1\right\}$. Then, the following holds true:*

$$\min_{\mathbf{z}\in\Delta_1}\ \sum_{i=1}^d a_i\mathbf{z}_i^r=\left(\sum_{i=1}^d a_i^{\frac{1}{1-r}}\right)^{1-r}$$

and the minimum is attained at

$$\mathbf{z}_i=a_i^{\frac{1}{1-r}}\left(\sum_{j=1}^d a_i^{\frac{1}{1-r}}\right)^{-1}\quad\forall\ i=1,\ldots,d.$$

Proof Take vectors $\mathbf{u}_1$ and $\mathbf{u}_2$ as those with entries $a_i^{\frac{1}{r}}\mathbf{z}_i$ and $a_i^{-\frac{1}{r}}\ \forall\ i=1,\ldots,d$ respectively. The result follows from the Holder's inequality: $\mathbf{u}_1^\top\mathbf{u}_2\leq\|\mathbf{u}_1\|_r\|\mathbf{u}_2\|_{\frac{r}{r-1}}$. Note that if any $a_i=0$, then the optimal value of the above optimization problem is zero. ■

Proposition 11 *The following convex optimization problems are dual to each other and there is no duality gap:*

$$\max_{\gamma\in\Delta_1}\ \sum_{w\in\mathcal{V}}\delta_w(\gamma,\lambda)M_w,\tag{18}$$

$$\min_{\kappa\in L}\ \max_{u\in\mathcal{V}}\ \sum_{w\in D(u)}\frac{\kappa_{uw}^2\lambda_{uw}M_w}{d_u^2},\tag{19}$$

where $L=\{\kappa\in\mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}\mid\kappa\geq 0,\ \sum_{v\in A(w)}\kappa_{vw}=1,\ \kappa_{vw}=0\ \forall v\in A(w)^c,\ \forall w\in\mathcal{V}\}$, $\Delta_1=\left\{\mathbf{z}\in\mathbb{R}^{|\mathcal{V}|}\mid\mathbf{z}\geq 0,\sum_{v\in\mathcal{V}}\mathbf{z}_v\leq 1\right\}$ and $M_w\geq 0\ \forall w\in\mathcal{V}$.

Proof The optimization problem (19) may be equivalently rewritten as:

$$\min_{\kappa\in L}\ \min_{A}A,\quad\text{subject to}\quad A\geq\sum_{w\in D(u)}\frac{\kappa_{uw}^2\lambda_{uw}M_w}{d_u^2}\ \forall u\in\mathcal{V},$$

$$= \min_{\kappa \in L} \max_{\gamma \in \Delta_1} \sum_{u \in \mathcal{V}} \sum_{w \in D(u)} \frac{\gamma_u \kappa_{uw}^2 \lambda_{uw} M_w}{d_u^2} \qquad \text{(Lagrangian dual with respect to } A\text{)}$$

$$= \max_{\gamma \in \Delta_1} \min_{\kappa \in L} \sum_{w \in \mathcal{V}} \left(\sum_{u \in A(w)} \kappa_{uw}^2 \frac{\gamma_u \lambda_{uw}}{d_u^2} \right) M_w \qquad \text{(min-max interchange and rearranging terms)}$$

$$= \max_{\gamma \in \Delta_1} \sum_{w \in \mathcal{V}} \left(\sum_{u \in A(w)} \left(\frac{\gamma_u \lambda_{uw}}{d_u^2} \right)^{-1} \right)^{-1} M_w \qquad \text{(Lemma 10 with respect to variables } \kappa\text{)}$$

$$= \max_{\gamma \in \Delta_1} \sum_{w \in \mathcal{V}} \delta_w(\gamma, \lambda) M_w \qquad \blacksquare$$

Lemma 12 *The following min-max interchange is equivalent:*

$$\min_{\gamma \in \Delta_1} \min_{\lambda_v \in \Delta_{\hat{\rho}}^v \ \forall v \in \mathcal{V}} \max_{\alpha_t \in S(\mathbf{y}_t, C) \forall t} G(\gamma, \lambda, \alpha) = \max_{\alpha_t \in S(\mathbf{y}_t, C) \forall t} \min_{\gamma \in \Delta_1} \min_{\lambda_v \in \Delta_{\hat{\rho}}^v \ \forall v \in \mathcal{V}} G(\gamma, \lambda, \alpha),$$

where $G(\gamma, \lambda, \alpha)$ is as defined in (10).

Proof We proceed by applying a change of variables. Note that $\gamma_v = 0$ implies that the variables λ_{vw} ($\forall w \in D(v)$) do not influence the objective of optimization problem (10). This follows from the definition of the $\delta(\gamma, \lambda)$ function. Hence, we define $\beta_{vw} = \gamma_v \lambda_{vw}$, $\forall w \in D(v)$ as it is a one-to-one transformation for $\gamma_v \neq 0$ (see also Szafranski et al., 2010). The gHKL dual (10) (the L.H.S. of the proposed lemma) can be equivalently rewritten as:

$$\min_{\substack{\beta_{vw} \geq 0 \ \forall w \in D(v), v \in \mathcal{V} \\ \sum_v \|\beta_{vD(v)}\|_{\hat{\rho}} \leq 1}} \max_{\alpha_t \in S(\mathbf{y}_t, C) \forall t} G(\beta, \alpha), \quad \text{where} \quad \beta_{vD(v)} = (\beta_{vw})_{w \in D(v)},$$

$$G(\beta, \alpha) = \mathbf{1}^\top \alpha - \frac{1}{2} \alpha^\top \mathbf{Y} \left(\sum_{w \in \mathcal{V}} \delta_w(\beta) H_w \right) \mathbf{Y} \alpha, \ \text{ and } \ \delta_w(\beta)^{-1} = \sum_{v \in A(w)} \frac{d_v^2}{\beta_{vw}}.$$

Note that $\delta_w(\beta)$ is a concave function of β (in the given feasibility set) and hence $G(\beta, \alpha)$ is convex-concave function with convex and compact feasibility sets. Therefore, we obtain $\min_\beta \max_\alpha G(\beta, \alpha) = \max_\alpha \min_\beta G(\beta, \alpha)$ (with constraints over β and α as stated above) by applying the Sion-Kakutani minimax theorem (Sion, 1958). Finally, we revert to the original variables (γ, λ) by substituting $\gamma_v = (\sum_{w \in D(v)} (\beta_{vw})^{\hat{\rho}})^{\frac{1}{\hat{\rho}}}$ $\forall v \in \mathcal{V}$ and $\lambda_{vw} = \frac{\beta_{vw}}{\gamma_v}$ $\forall w \in D(v)$, $\forall v \in \mathcal{V}$ s.t. $\gamma_v \neq 0$. This gives us the equivalent R.H.S. $\blacksquare$

Now we begin the proof of Theorem 3.

Proof From Lemma 12, the gHKL dual (10) can be equivalently written as:

$$\max_{\alpha \in S(\mathbf{y}, C)} \mathbf{1}^\top \alpha - \frac{1}{2} \underbrace{\max_{\gamma \in \Delta_1} \max_{\lambda_v \in \Delta_{\hat{\rho}}^v \ \forall v \in \mathcal{V}} \left(\sum_{w \in \mathcal{V}} \delta_w(\gamma, \lambda) \alpha^\top \mathbf{Y} H_w \mathbf{Y} \alpha \right)}_{\mathcal{O}}, \tag{20}$$

where $\hat{\rho} = \frac{\rho}{2-\rho}$. In the following, we equivalently rewrite the second part of the above formulation.

$$
\begin{aligned}
\mathcal{O} &= \max_{\lambda_v \in \Delta_{\hat{\rho}}^v \ \forall v \in \mathcal{V}} \ \max_{\gamma \in \Delta_1} \sum_{w \in \mathcal{V}} \delta_w(\gamma, \lambda) \underbrace{\alpha^\top \mathbf{Y} H_w \mathbf{Y} \alpha}_{M_w} \\
&= \max_{\lambda_v \in \Delta_{\hat{\rho}}^v \ \forall v \in \mathcal{V}} \ \min_{\kappa \in L} \ \max_{u \in \mathcal{V}} \sum_{w \in D(u)} \frac{\kappa_{uw}^2 \lambda_{uw} M_w}{d_u^2} && \text{(Proposition 11)} \\
&= \max_{\lambda_v \in \Delta_{\hat{\rho}}^v \ \forall v \in \mathcal{V}} \ \min_{\kappa \in L} \ \min_{A} \ A && \text{(Eliminating } u\text{)} \\
&\qquad \text{s.t. } A \geq \sum_{w \in D(v)} \frac{\kappa_{vw}^2 \lambda_{vw} M_w}{d_v^2} \ \forall v \in \mathcal{V} \\
&= \min_{\kappa \in L} \ \min_{A} \ \max_{\lambda_v \in \Delta_{\hat{\rho}}^v \ \forall v \in \mathcal{V}} \ A && \text{(Sion-Kakutani theorem)} \\
&\qquad \text{s.t. } A \geq \sum_{w \in D(v)} \frac{\kappa_{vw}^2 \lambda_{vw} M_w}{d_v^2} \ \forall v \in \mathcal{V} \\
&= \min_{\kappa \in L} \ \min_{A} \ A && \text{(Holder's inequality, } \bar{\rho} = \tfrac{\hat{\rho}}{\hat{\rho}-1}\text{)} \\
&\qquad \text{s.t. } A \geq d_v^{-2} \left(\sum_{w \in D(v)} \left(\kappa_{vw}^2 M_w \right)^{\bar{\rho}} \right)^{\frac{1}{\bar{\rho}}} \ \forall v \in \mathcal{V} \\
&= \min_{\kappa \in L} \ \max_{u \in \mathcal{V}} \ d_u^{-2} \left(\sum_{w \in D(u)} \left(\kappa_{uw}^2 M_w \right)^{\bar{\rho}} \right)^{\frac{1}{\bar{\rho}}} && \text{(Eliminating A)} \qquad (21)
\end{aligned}
$$

Now consider the problem $\mathcal{O}^{\bar{\rho}} = \min_{\kappa \in L} \max_{u \in \mathcal{V}} d_u^{-2\bar{\rho}} \sum_{w \in D(u)} \left(\kappa_{uw}^2 M_w \right)^{\bar{\rho}}$. Its Lagrangian is

$$
\mathcal{L}(\kappa, A, \eta) = A + \sum_{v \in \mathcal{V}} \eta_v \left(d_v^{-2\bar{\rho}} \sum_{w \in D(v)} \left(\kappa_{vw}^2 M_w \right)^{\bar{\rho}} - A \right).
$$

Minimization of $\mathcal{L}$ with respect to A leads to the constraint $\eta \in \Delta_1$. Hence, we have:

$$
\mathcal{O}^{\bar{\rho}} = \max_{\eta \in \Delta_1} \ \min_{\kappa \in L} \ \sum_{v \in \mathcal{V}} \sum_{w \in D(v)} \eta_v \left(d_v^{-2} \kappa_{vw}^2 M_w \right)^{\bar{\rho}}.
$$

Using the special structure of L, the above can be rewritten as:

$$
\mathcal{O}^{\bar{\rho}} = \max_{\eta \in \Delta_1} \ \sum_{w \in \mathcal{V}} (M_w)^{\bar{\rho}} \left(\min_{\kappa_w \in \Delta_{|A(w)|}} \sum_{v \in A(w)} \left(\eta_v d_v^{-2\bar{\rho}} \right) \kappa_{vw}^{2\bar{\rho}} \right),
$$

where $\Delta_{|A(w)|} = \left\{ \eta \in \mathbb{R}^{|A(w)|} \mid \eta \geq 0, \sum_{w \in A(w)} \eta_w \leq 1 \right\}$. By applying Lemma 10 with respect to variables κ, we obtain the following equivalence:

$$
\min_{\kappa_w \in \Delta_{|A(w)|}} \sum_{v \in A(w)} \left(\eta_v d_v^{-2\bar{\rho}} \right) \kappa_{vw}^{2\bar{\rho}} = \zeta_w(\eta) = \left(\sum_{v \in A(w)} d_v^{\rho} \eta_v^{1-\rho} \right)^{\frac{1}{1-\rho}}. \qquad (22)
$$

From the above two results, we obtain the following equivalent dual of (21):

$$\mathcal{O} = \max_{\eta \in \Delta_1} \left(\sum_{w \in \mathcal{V}} \zeta_w(\eta) \left(\alpha^\top \mathbf{Y} H_w \mathbf{Y} \alpha \right)^{\bar{\rho}} \right)^{\frac{1}{\bar{\rho}}}. \tag{23}$$

Substituting $\mathcal{O}$ in (20) by the above (23) and again interchanging the min-max completes the proof. ■

A.6 Proof of Theorem 4

Proof We begin by noting that $\zeta_v(\eta)$ $(v \in \mathcal{V})$ is a concave function of η for all v (this is because when $\rho \in (1, 2]$, ζ_v is a weighted q-norm in η, where $q \in [-1, 0)$ and hence is concave in the first quadrant). By simple observations regarding operations preserving convexity we have that the objective in (13) is a convex function of η for a fixed value of α. Hence $g(\eta)$, which is a point-wise maximum over convex functions, is itself convex. The expression for $\nabla g(\eta)$ is computed by employing Danskin's theorem (Bertsekas, 1999, Proposition B.25) and is as follows:

$$(\nabla g(\eta))_i = -\frac{(1-\varepsilon)}{2\bar{\rho}} \times \overbrace{\left(\sum_{u \in D(i)} d_i^\rho \left((1-\varepsilon)\eta_i + \frac{\varepsilon}{|\mathcal{V}|} \right)^{-\rho} \zeta_u^s(\eta)^\rho \left(\bar{\alpha}^\top \mathbf{Y} H_u \mathbf{Y} \bar{\alpha} \right)^{\bar{\rho}} \right)}^{P_1} \tag{24}$$
$$\times \underbrace{\left(\sum_{w \in \mathcal{V}} \zeta_w^s(\eta) \left(\bar{\alpha}^\top \mathbf{Y} H_w \mathbf{Y} \bar{\alpha} \right)^{\bar{\rho}} \right)^{\frac{1}{\bar{\rho}}-1}}_{P_2},$$

where $\bar{\rho} = \frac{\rho}{2(\rho-1)}$, $\zeta_w^s(\eta) = \zeta_w((1-\varepsilon)\eta + \frac{\varepsilon}{|\mathcal{V}|})$, i.e., the smoothed $\zeta_w(\eta)$ and $\bar{\alpha}$ is an optimal solution of problem (13) with that η where the gradient is to be computed.

Next, we show that g is Lipschitz continuous by showing that its gradient is bounded. Firstly, $\rho \in (1, 2]$ and hence $\bar{\rho} \in [1, \infty)$. Next, let the minimum and maximum eigenvalues over all H_w $(w \in \mathcal{V})$ be θ and σ respectively. Then we have $\theta \|\bar{\alpha}\|^2 \leq \bar{\alpha}^\top \mathbf{Y} H_w \mathbf{Y} \bar{\alpha} \leq \sigma \|\bar{\alpha}\|^2$. Using this, we obtain: $\sum_{w \in \mathcal{V}} \zeta_w^s(\eta) \left(\bar{\alpha}^\top \mathbf{Y} H_w \mathbf{Y} \bar{\alpha} \right)^{\bar{\rho}} \geq \theta^{\bar{\rho}} \|\bar{\alpha}\|^{2\bar{\rho}} \sum_{w \in \mathcal{V}} \zeta_w^s(\eta)$. Note that $\sum_{w \in \mathcal{V}} \zeta_w^s(\eta) \geq \zeta_r^s(\eta)$ where $r \in sources(\mathcal{V})$ and $\zeta_r^s(\eta) \geq d_{max}^{\rho/(1-\rho)} \frac{\varepsilon}{|\mathcal{V}|}$ where d_{max} is the maximum of d_v $(v \in \mathcal{V})$. Thus we obtain: $P_2 \leq \left(\theta^{\bar{\rho}} \|\bar{\alpha}\|^{2\bar{\rho}} \varepsilon / |\mathcal{V}| \right)^{\frac{1}{\bar{\rho}}-1} d_{max}^{\frac{2-\rho}{\rho-1}}$.

Now, it is easy to see that $\forall u \in D(i)$, $d_i^\rho ((1-\varepsilon)\eta_i + \frac{\varepsilon}{|\mathcal{V}|})^{-\rho} \zeta_u(\eta)^\rho \leq d_i^{\frac{\rho}{1-\rho}} \leq d_{min}^{\frac{\rho}{1-\rho}}$, where d_{min} is the minimum of d_v $(v \in \mathcal{V})$. Hence $P_1 \leq |\mathcal{V}| \sigma^{\bar{\rho}} \|\bar{\alpha}\|^{2\bar{\rho}} d_{min}^{\frac{\rho}{1-\rho}}$. In addition, since $0 \leq \bar{\alpha} \leq C$, we have $\|\bar{\alpha}\| \leq \sqrt{mT}C$. Summarizing these findings, we obtain the following bound on the gradient:

$$\|\nabla g(\eta)\|_1 \leq \frac{(1-\varepsilon)}{2\bar{\rho}} mTC^2 \theta^{1-\bar{\rho}} \sigma^{\bar{\rho}} \varepsilon^{\frac{1-\bar{\rho}}{\bar{\rho}}} |\mathcal{V}|^{\frac{2}{\rho}+1} d_{min}^{\frac{\rho}{1-\rho}} d_{max}^{\frac{2-\rho}{\rho-1}}.$$

The proof will be similar for gHKL$_{\text{MT}}$ formulations in other learning settings. ■

A.7 Proof of Theorem 5

Proof Given a candidate solution η and $\alpha = [\alpha_1^\top, \ldots, \alpha_T^\top]^\top$ (with associated primal ($\mathbf{f} = (f_1, \ldots, f_T), b, \xi$)), the duality gap ($D$) between the two variational formulations in Lemma 9 is as follows:

$$\begin{aligned} D &= \max_{\hat{\alpha}_t \in S(\mathbf{y}_t, C) \forall t} \bar{G}(\eta, \hat{\alpha}) \; - \min_{\hat{\eta} \in \Delta_1} \; \bar{G}(\hat{\eta}, \alpha) \\ &\leq \frac{1}{2}\Omega_T(\mathbf{f})^2 + C\mathbf{1}^\top \xi \; - \min_{\hat{\eta} \in \Delta_1} \; \bar{G}(\hat{\eta}, \alpha) \\ &= \overbrace{\Omega_T(\mathbf{f})^2 + C\mathbf{1}^\top \xi - \mathbf{1}^\top \alpha}^{\text{Gap in solving with fixed } \eta} + \underbrace{\frac{1}{2}\left(\max_{\hat{\eta} \in \Delta_1} \; \left(\sum_{w \in \mathcal{V}} \zeta_w(\hat{\eta}) \left(\alpha^\top \mathbf{Y} H_w \mathbf{Y} \alpha \right)^{\bar{\rho}} \right)^{\frac{1}{\bar{\rho}}} - \Omega_T(\mathbf{f})^2 \right)}_{\text{Gap in solving with fixed } \alpha}. \end{aligned}$$

With this upper bound on the duality gap, it is easy to see that the following condition is sufficient for the reduced solution (with active set $\mathcal{W}$) to have $D \leq \epsilon$:

$$\max_{\eta \in \Delta_1} \; \left(\sum_{w \in \mathcal{V}} \zeta_w(\eta) \left(\alpha^\top \mathbf{Y} H_w \mathbf{Y} \alpha \right)^{\bar{\rho}} \right)^{\frac{1}{\bar{\rho}}} \leq \Omega_T(\mathbf{f}_{\mathcal{W}})^2 + 2(\epsilon - \epsilon_{\mathcal{W}}), \tag{25}$$

where $\epsilon_{\mathcal{W}}$ is the duality gap[9] associated with the computation of the dual variables $\alpha_{\mathcal{W}}$. Here as well as in the rest of the proof, the subscript $(\cdot)_{\mathcal{W}}$ implies the value of the variable obtained when the gHKL$_{\text{MT}}$ formulation is solved with $\mathcal{V}$ restricted to the active set $\mathcal{W}$. In Appendix A.5, we had proved that the L.H.S. of the above inequality is equal to the R.H.S. of (21), i.e.,

$$\max_{\eta \in \Delta_1} \; \left(\sum_{w \in \mathcal{V}} \zeta_w(\eta) \left(M_w \right)^{\bar{\rho}} \right)^{\frac{1}{\bar{\rho}}} = \min_{\kappa \in L} \; \max_{v \in \mathcal{V}} \; d_v^{-2} \left(\sum_{w \in D(v)} \left(\kappa_{vw}^2 M_w \right)^{\bar{\rho}} \right)^{\frac{1}{\bar{\rho}}}, \tag{26}$$

where $M_w = \alpha_{\mathcal{W}}^\top \mathbf{Y} H_w \mathbf{Y} \alpha_{\mathcal{W}}$.

Next, we obtain an upper bound of the above by substituting $\kappa \in L$ in the R.H.S of (26). In particular, we employ the following: the value of κ_{vw} $v, w \in \mathcal{W}$ is obtained by solving the small[10] problem (14). This is fine because $\mathcal{W} = hull(\mathcal{W})$. For $v \in \mathcal{W}^c$ and $w \in \mathcal{W}$, by definition of L and $\mathcal{W}$, we have $\kappa_{vw} = 0$. Next, κ_{vw} is set to zero $\forall$ $v \in \mathcal{W}, w \in \mathcal{W}^c$. For the remaining κ_{vw}, $v \in \mathcal{W}^c$ and $w \in \mathcal{W}^c$, we use the value of κ obtained by solving (21) with $\rho = 1$, i.e., $\kappa_{vw} = d_v \left(\sum_{u \in A(v) \cap \mathcal{W}^c} d_u \right)^{-1}$ (also see Section A.5 Bach, 2009). Note that the above constructed value of κ is feasible in the set L. With this choice of κ substituted in

9. This is given by the gap associated with the $\hat{\rho}$-norm MKL solver employed in the mirror descent algorithm for solving the small problem (14).
10. The value of κ_{vw} ($\forall v, w \in \mathcal{W}$) obtained in this manner satisfy the constraint set L restricted to $\mathcal{W}$, i.e., $L_{\mathcal{W}} = \{\kappa \in \mathbb{R}^{|\mathcal{W}| \times |\mathcal{W}|} \mid \kappa \geq 0, \; \sum_{v \in A(w)} \kappa_{vw} = 1, \; \kappa_{vw} = 0 \; \forall \; v \in A(w)^c \cap \mathcal{W}, \; \forall \; w \in \mathcal{W}\}$

the R.H.S. of (26), we have the following inequalities:

$$\begin{aligned}
&\max_{\eta\in\Delta_1} \left(\sum_{w\in\mathcal{V}} \zeta_w(\eta)\left(\alpha_{\mathcal{W}}^\top \mathbf{Y} H_w \mathbf{Y} \alpha_{\mathcal{W}}\right)^{\bar\rho}\right)^{\frac{1}{\bar\rho}} \\
&\quad \leq \max\left\{\Omega_T(\mathbf{f}_{\mathcal{W}})^2, \max_{u\in\mathcal{W}^c}\left(\sum_{w\in D(u)}\left(\frac{\alpha_{\mathcal{W}}^\top \mathbf{Y} H_w \mathbf{Y} \alpha_{\mathcal{W}}}{\left(\sum_{v\in A(w)\cap\mathcal{W}^c} d_v\right)^2}\right)^{\bar\rho}\right)^{\frac{1}{\bar\rho}}\right\} \\
&\qquad (\text{ Specific choice of } \kappa) \\
&\quad = \max\left\{\Omega_T(\mathbf{f}_{\mathcal{W}})^2, \max_{u\in sources(\mathcal{W}^c)}\left(\sum_{w\in D(u)}\left(\frac{\alpha_{\mathcal{W}}^\top \mathbf{Y} H_w \mathbf{Y} \alpha_{\mathcal{W}}}{\left(\sum_{v\in A(w)\cap\mathcal{W}^c} d_v\right)^2}\right)^{\bar\rho}\right)^{\frac{1}{\bar\rho}}\right\} \\
&\qquad (\because\ \mathcal{W} = hull(\mathcal{W})) \\
&\quad \leq \max\left\{\Omega_T(\mathbf{f}_{\mathcal{W}})^2, \max_{u\in sources(\mathcal{W}^c)}\left(\sum_{w\in D(u)}\left(\frac{\alpha_{\mathcal{W}}^\top \mathbf{Y} H_w \mathbf{Y} \alpha_{\mathcal{W}}}{\left(\sum_{v\in A(w)\cap D(u)} d_v\right)^2}\right)^{\bar\rho}\right)^{\frac{1}{\bar\rho}}\right\} \\
&\qquad (\because\ \textstyle\sum_{v\in A(w)\cap\mathcal{W}^c} d_v \geq \sum_{v\in A(w)\cap D(u)} d_v) \\
&\quad \leq \max\left\{\Omega_T(\mathbf{f}_{\mathcal{W}})^2, \max_{u\in sources(\mathcal{W}^c)}\sum_{w\in D(u)}\frac{\alpha_{\mathcal{W}}^\top \mathbf{Y} H_w \mathbf{Y} \alpha_{\mathcal{W}}}{\left(\sum_{v\in A(w)\cap D(u)} d_v\right)^2}\right\} \\
&\qquad (\because\ \|\beta\|_1 \geq \|\beta\|_{\bar\rho}\ \forall\ \bar\rho \geq 1)
\end{aligned}$$

Employing the above upper bound in (25) leads to the result in Theorem 5. Note that in practice, the last upper bound is not loose for Rule Ensemble Learning (REL) application. This is because most of the matrices, especially near the bottom of the lattice, will be (near) zero-matrices — larger the conjunctive rule, the fewer are the examples which may satisfy it. ■

A.8 gHKL$_{\mathrm{MT}}$ with General Convex Loss Functions

In this section, we present extension of the proposed algorithm to other learning settings like regression. In particular, we consider the case where the loss function $\ell(\cdot,\cdot)$ is a general convex loss function such as the hinge loss, the square loss, the Huber loss, etc.

The gHKL$_{\mathrm{MT}}$ primal formulation with a general convex loss function $\ell(\cdot,\cdot)$ was given in equation (6). The specialized gHKL$_{\mathrm{MT}}$ dual formulation corresponding to (6) is as follows:

$$\min_{\eta\in\Delta_1}\ \max_{\alpha_t\in\mathbb{R}^m, \mathbf{1}^\top\alpha_t=0\ \forall t}\ -C\sum_{t=1}^{T}\sum_{i=1}^{m}\varphi_{ti}^*\left(-\frac{\alpha_{ti}}{C}\right) - \frac{1}{2}\left(\sum_{w\in\mathcal{V}}\zeta_w(\eta)\left(\alpha^\top H_w \alpha\right)^{\bar\rho}\right)^{\frac{1}{\bar\rho}},$$

where $\alpha = [\alpha_1^\top, \ldots, \alpha_T^\top]^\top$, $\zeta_w(\eta) = \left(\sum_{v \in A(w)} d_v^\rho \eta_v^{1-\rho}\right)^{\frac{1}{1-\rho}}$ (refer Theorem 3 for details) and φ_{ti}^* denotes the Fenchel[11] conjugate (Boyd and Vandenberghe, 2004) of the function $\varphi_{ti} : z \to \ell(y_{ti}, z)$.

A.9 Prediction Function for gHKL$_{\mathbf{MT}}$ with the Hinge Loss Function

Let the final active set be $\mathcal{W}$ and $(\bar{\eta}_\mathcal{W}, \bar{\alpha}_\mathcal{W})$ be the optimal solution of (12). Then the prediction function for an instance $\mathbf{x}_{tj}$ belonging to the t^{th} task is given by

$$F_t(\mathbf{x}) = (\bar{\alpha}_\mathcal{W} \odot \mathbf{y})^\top \left(\sum_{w \in \mathcal{W}} \bar{\theta}_w (\zeta_w(\bar{\eta}_\mathcal{W}))^{\frac{1}{\bar{\rho}}} H_w(\cdot, \mathbf{x}_{tj}) \right), \tag{27}$$

where symbol $\odot$ denote element-wise product, H_w is the kernel matrix corresponding to the multi-task kernel (11), $H_w(\cdot, \mathbf{x}_{tj}) = ((H_w(\mathbf{x}_{t'i}, \mathbf{x}_{tj}))_{i=1}^m)_{t'=1}^T$ and

$$\bar{\theta}_w = \left(\frac{(\zeta_w(\bar{\eta}_\mathcal{W}))^{\frac{1}{\bar{\rho}}} \bar{\alpha}_\mathcal{W}^\top \mathbf{Y} H_w \mathbf{Y} \bar{\alpha}_\mathcal{W}}{\left(\sum_{v \in \mathcal{W}} \left((\zeta_v(\bar{\eta}_\mathcal{W}))^{\frac{1}{\bar{\rho}}} \bar{\alpha}_\mathcal{W}^\top \mathbf{Y} H_v \mathbf{Y} \bar{\alpha}_\mathcal{W} \right)^{\bar{\rho}} \right)^{\frac{1}{\bar{\rho}}}} \right)^{\frac{1}{\bar{\rho}-1}}.$$

A.10 Proof of Corollary 6

Note that proving the computational complexity of the matrix $\mathcal{K}_u$ ($u \in \mathit{sources}(\mathcal{W}^c)$) in (15) to be polynomial time in size of the active set and the training set dimensions suffices to prove the corollary. This is because all the other steps in Algorithms 3 and 2 are of polynomial time complexity (discussed in Section 4).

We begin the proof by introducing some indexing notations related to the multi-task matrices. Let the entries in H_w, the $mT \times mT$ multi-task kernel matrix, be arranged in the following form: the entry corresponding to the input pair $(\mathbf{x}_{t_1 i}, \mathbf{x}_{t_2 j})$ be in the $((t_1 - 1) * m + i)^{th}$ row and $((t_2 - 1) * m + j)^{th}$ column of H_w.

Next we observe that the expression for $\mathcal{K}_u$ in Theorem 5 may be rewritten as:

$$\mathcal{K}_u = \underbrace{\left(\sum_{w \in D(u)} \frac{K_w}{\left(\sum_{v \in A(w) \cap D(u)} d_v \right)^2} \right)}_{T_u} \odot K_T,$$

where: i) K_w is a $mT \times mT$ matrix corresponding to the base kernel k_w and constructed from the inputs from all the tasks, ii) K_T is a $mT \times mT$ such that the entry corresponding to the $((t_1 - 1) * m + i)^{th}$ row and $((t_2 - 1) * m + j)^{th}$ column ($1 \leq i, j \leq m$) of K_T is $B(t_1, t_2)$, and iii) $\odot$ is the symbol for element-wise product (Hadamard product).

11. Fenchel conjugate $\varphi^*(z)$ of a convex function $\varphi(u)$ is given by $\varphi^*(z) = \sup_u z^\top u - \varphi(u)$. As an example, for hinge loss $\varphi(u) = \ell(u, y) = \max(0, 1 - uy)$, $\varphi^*(z) = \begin{cases} zy & \text{if } zy \in [-1, 0] \\ \infty & \text{otherwise} \end{cases}$

In the above expression, $\mathcal{K}_u$ is computable in polynomial time if and only if T_u is computable in polynomial time. The proof of the corollary follows from observing the expression of the sufficient condition for optimality of the HKL (Bach, 2009, Equation 21), which also involves the term T_u.

A.11 Proof of Theorem 7

Given an active set $\mathcal{W}$ of size W, proving that the computational complexity of the verification of the sufficient condition of optimality (15) is polynomial in terms of the active set and the training set sizes suffices to prove Theorem 7. This is because all the other steps in Algorithms 3 and 2 are of polynomial time complexity (discussed in Section 4).

In the REL setup, the DAG is the conjunction lattice and the embedded kernels k_v $v \in \mathcal{V}$ may be rewritten as:

$$k_v(\mathbf{x}_i, \mathbf{x}_j) = \phi_v(\mathbf{x}_i) \cdot \phi_v(\mathbf{x}_j) = \left(\prod_{c \in S_v} \phi_c(\mathbf{x}_i) \right) \cdot \left(\prod_{c \in S_v} \phi_c(\mathbf{x}_j) \right) = \bigodot_{c \in S_v} k_c(\mathbf{x}_i, \mathbf{x}_j),$$

where S_v is the set of basic propositions involved in the conjunction ϕ_v and $\odot$ is the symbol for element-wise product (Hadamard product). The kernels corresponding to the basic propositions are in fact the base kernels embedded in the second level nodes of the lattice $\mathcal{V}$. Employing the above definition of $k_v(\mathbf{x}_i, \mathbf{x}_j)$, the matrices $\mathcal{K}_u$ (in L.H.S. of (15)) are computed as:

$$\mathcal{K}_u = \sum_{w \in D(u)} \frac{K_w}{\left(\sum_{v \in A(w) \cap D(u)} d_v \right)^2} = \left(\bigodot_{c \in S_u} \frac{K_c}{a^2} \right) \odot \left(\bigodot_{c \in B/S_u} \left(\frac{K_c}{(1+a)^2} + \mathbf{1}\mathbf{1}^\top \right) \right),$$

where K_c is the kernel matrix corresponding to the basic proposition ϕ_c, B is the set of all basic propositions and the parameters d_v ($v \in \mathcal{V}$) are defined as $d_v = a^{|S_v|}$ ($a > 0$).

It is obvious that a trivial computational complexity of computing $\mathcal{K}_u$ ($u \in \mathcal{V}$) is $O(pm^2)$. In practice, this complexity can be reduced to $O(m^2)$ by caching the matrices $\mathcal{K}_u$. For illustration, suppose $\mathcal{K}_{u_1}$ needs to be computed, given that $\mathcal{K}_{u_0}$ is cached and u_0 is a parent of u_1. Let the extra basic proposition contained in ϕ_{u_1} (with respect to ϕ_{u_0}) be ϕ_e. Then $\mathcal{K}_{u_1}$ can be calculated as follows:

$$\mathcal{K}_{u_1} = \mathcal{K}_{u_0} \odot \left(\frac{K_e}{a^2} \right) \oslash \left(\frac{K_e}{(1+a)^2} + \mathbf{1}\mathbf{1}^\top \right),$$

where $\oslash$ is the symbol for element-wise division of matrices.

Hence, plugging the REL specific values in the runtime complexity of the gHKL algorithm, ω =constant and $z = p$, the runtime complexity of the gHKL based REL algorithm is $O(m^3W^3 \log(W) + m^2W^2p)$.

A.12 REL Binary Classification Results in AUC

Table 5 reports the REL binary classification results in AUC (area under the ROC curve). The experimental details (and results measured in F1-score) are discussed in Section 6.

	RuleFit	**SLI**	**ENDER**	**HKL-ℓ_1-MKL**	**gHKL$_\rho$**		
					$\rho = 2$	$\rho = 1.5$	$\rho = 1.1$
TIC	0.736 ± 0.05	0.482 ± 0.21	0.783 ± 0.036	0.836 ± 0.024	0.967 ± 0.023	0.973 ± 0.02	$\mathbf{0.975 \pm 0.018}$
BCW	0.941 ± 0.011	0.917 ± 0.051	0.958 ± 0.039	0.981 ± 0.008	$\mathbf{0.984 \pm 0.005}$	0.93 ± 0.099	0.93 ± 0.099
DIA	0.67 ± 0.027	0.576 ± 0.115	0.761 ± 0.02	0.746 ± 0.050	$\mathbf{0.766 \pm 0.046}$	0.733 ± 0.058	0.636 ± 0.118
HAB	0.537 ± 0.054	0.17 ± 0.155	$\mathbf{0.575 \pm 0.039}$	0.524 ± 0.078	0.556 ± 0.07	0.482 ± 0.11	0.383 ± 0.166
HTC	0.764 ± 0.03	0.541 ± 0.215	0.805 ± 0.031	0.802 ± 0.085	$\mathbf{0.837 \pm 0.035}$	0.763 ± 0.12	0.753 ± 0.118
BLD	0.546 ± 0.06	0.175 ± 0.256	$\mathbf{0.68 \pm 0.028}$	0.660 ± 0.025	0.667 ± 0.034	0.634 ± 0.028	0.519 ± 0.079
HTS	0.765 ± 0.028	0.712 ± 0.085	0.801 ± 0.022	0.825 ± 0.032	$\mathbf{0.849 \pm 0.021}$	0.83 ± 0.027	0.811 ± 0.056
MK3	0.972	0.632	0.998	0.995	**1**	0.998	0.957
VTE	0.955 ± 0.022	0.919 ± 0.048	0.965 ± 0.014	$\mathbf{0.977 \pm 0.009}$	0.972 ± 0.016	0.948 ± 0.015	0.945 ± 0.016
BCC	0.578 ± 0.05	0.469 ± 0.078	0.622 ± 0.043	0.627 ± 0.063	$\mathbf{0.637 \pm 0.055}$	0.576 ± 0.089	0.513 ± 0.124
MAM	0.818 ± 0.02	0.763 ± 0.08	$\mathbf{0.887 \pm 0.006}$	0.866 ± 0.028	0.882 ± 0.023	0.85 ± 0.032	0.839 ± 0.03
LIV	0.607 ± 0.017	0.093 ± 0.168	0.619 ± 0.038	0.619 ± 0.074	$\mathbf{0.623 \pm 0.038}$	0.583 ± 0.11	0.565 ± 0.109

Table 5: Results on binary REL classification. We report the average AUC along with standard deviation, over ten random train-test splits.

References

J. Aflalo, A. Ben-Tal, C. Bhattacharyya, J. Saketha Nath, and S. Raman. Variable sparsity kernel learning. *Journal of Machine Learning Research*, 12:565–592, 2011.

M. A. Álvarez, L. Rosasco, and N. D. Lawrence. Kernels for vector-valued functions: a review. *Foundations and Trends in Machine Learning*, 4:195–266, 2012.

R. K. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853, 2005.

A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine Learning*, 73:243–272, 2008.

F. Bach. Exploring large feature spaces with hierarchical multiple kernel learning. In *Advances in Neural Information Processing Systems*, 2008.

F. Bach. High-dimensional non-linear variable selection through hierarchical kernel learning. Technical report, INRIA, France, 2009.

F. Bach, G. R. G. Lanckriet, and M. I. Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. In *Proceedings of International Conference on Machine Learning*, 2004.

J. Baxter. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12:149–198, 2000.

A. Beck and M. Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

S. Ben-David and R. Schuller. A notion of task relatedness yielding provable multiple-task learning guarantees. *Machine Learning*, 73:273–287, 2008.

A. Ben-Tal and A. Nemirovski. Lectures on modern convex optimization: Analysis, algorithms and engineering applications. *MPS/ SIAM Series on Optimization*, 1, 2001.

D. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

K. Blake and M. Lichman. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml`.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

R. Caruana. Mutitask learning. *Machine Learning*, 28:41–75, 1997.

P. Clark and T. Niblett. The CN2 induction algorithm. *Machine Learning*, 3:261–283, 1989.

W. W. Cohen and Y. Singer. A simple, fast, and effective rule learner. In *AAAI Conference on Artificial Intelligence*, 1999.

T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms (3. ed.)*. MIT Press, 2009.

K. Dembczyński, W. Kotłowski, and R. Słowiński. Maximum likelihood rule ensembles. In *Proceedings of the International Conference of Machine Learning*, 2008.

K. Dembczyński, W. Kotłowski, and R. Słowiński. ENDER - A statistical framework for boosting decision rules. *Data Mining and Knowledge Discovery*, 21:52–90, 2010.

T. Evgeniou and M. Pontil. Regularized multi-task learning. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004.

T. Evgeniou, C. A. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6:615–637, 2005.

J. H. Friedman and B. E. Popescu. Predictive learning via rule ensembles. *Annals of Applied Statistics*, 2:916–954, 2008.

L. Jacob, F. Bach, and J.-P. Vert. Clustered multi-task learning: A convex formulation. In *Advances in Neural Information Processing Systems*, 2008.

A. Jain, S. V. N. Vishwanathan, and M. Varma. SPG-GMKL: Generalized multiple kernel learning with a million kernels. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2012.

P. Jawanpuria and J. S. Nath. Multi-task multiple kernel learning. In *SIAM International Conference on Data Mining*, 2011.

P. Jawanpuria and J. S. Nath. A convex feature learning formulation for latent task structure discovery. In *Proceedings of the International Conference on Machine Learning*, 2012.

P. Jawanpuria, J. S. Nath, and G. Ramakrishnan. Efficient rule ensemble learning using hierarchical kernels. In *Proceedings of the International Conference of Machine Learning*, 2011.

P. Jawanpuria, M. Varma, and J. S. Nath. On p-norm path following in multiple kernel learning for non-linear feature selection. In *Proceedings of the International Conference of Machine Learning*, 2014.

M. Kloft, U. Brefeld, S. Sonnenburg, and A. Zien. ℓ_p-norm multiple kernel learning. *Journal of Maching Learning Research*, 12:953–997, 2011.

G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. I. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5: 27–72, 2004.

H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *Advances in Neural Information Processing Systems*, 2007.

J. Liu and J. Ye. Efficient ℓ_1/ℓ_q norm regularization. Technical Report arXiv:1009.4766, 2010.

K. Lounici, M. Pontil, A. B. Tsybakov, and S. van de Geer. Taking advantage of sparsity in multi-task learning. In *Proceedings of the Annual Conference on Learning Theory*, 2009.

C. A. Micchelli and M. Pontil. Learning the kernel function via regularization. *Journal of Machine Learning Research*, 6:1099–1125, 2005.

R. S. Michalski. A theory and methodology of inductive learning. *Artificial Intelligence*, 20:111–161, 1983.

S. Negahban and M. Wainwright. Phase transitions for high-dimensional joint support recovery. In *Advances in Neural Information Processing Systems*, 2009.

G. Obozinski, Martin J. Wainwright, and M.I. Jordan. Support union recovery in high-dimensional multivariate regression. *Annals of Statistics*, 39:1–17, 2011.

F. Orabona, J. Luo, and B. Caputo. Multi kernel learning with online-batch optimization. *Journal of Machine Learning Research*, 13:227–253, 2012.

J. C. Platt. Fast training of support vector machines using sequential minimal optimization. In *Advances in Kernel Methods - Support Vector Learning*, 1999.

J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet. SimpleMKL. *Journal of Machine Learning Research*, 9:2491–2521, 2008.

R. L. Rivest. Learning decision lists. *Machine Learning*, 2:229–246, 1987.

B. Schölkopf and A. Smola. *Learning with Kernels*. MIT press, Cambridge, 2002.

D. Sheldon. Graphical multi-task learning. Technical report, Cornell University, 2008.

M. Sion. On general minimax theorem. *Pacific Journal of Mathematics*, 1958.

M. Szafranski, Y. Grandvalet, and P. M. Mahoudeaux. Hierarchical penalization. In *Advances in Neural Information Processing Systems*, 2007.

M. Szafranski, Y. Grandvalet, and A. Rakotomamonjy. Composite kernel learning. *Machine Learning*, 79:73–103, 2010.

B. A. Turlach, W. N. Venables, and S. J. Wright. Simultaneous variable selection. *Technometrics*, 47:349–363, 2005.

V. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.

S. V. N. Vishwanathan, Z. Sun, N. T.-Ampornpunt, and M. Varma. Multiple kernel learning and the SMO algorithm. In *Advances in Neural Information Processing Systems*, 2010.

S. M. Weiss and N. Indurkhya. Lightweight rule induction. In *Proceedings of the International Conference of Machine Learning*, 2000.

Y. Xue, X. Liao, L. Carin, and B. Krishnapuram. Multi-task learning for classification with dirichlet process priors. *Journal of Maching Learning Research*, 8:35–63, 2007.

M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68:49–67, 2006.

Journal of Machine Learning Research 16 (2015) 653-672 Submitted 6/13; Published 4/15

Discrete Restricted Boltzmann Machines

Guido Montúfar MONTUFAR@MIS.MPG.DE
Max Planck Institute for Mathematics in the Sciences
Inselstrasse 22, 04103 Leipzig, Germany
Department of Mathematics
Pennsylvania State University
University Park, PA 16802, USA

Jason Morton MORTON@MATH.PSU.EDU
Department of Mathematics
Pennsylvania State University
University Park, PA 16802, USA

Editor: Aaron Courville, Rob Fergus, and Christopher Manning

Abstract

We describe discrete restricted Boltzmann machines: probabilistic graphical models with bipartite interactions between visible and hidden discrete variables. Examples are binary restricted Boltzmann machines and discrete naïve Bayes models. We detail the inference functions and distributed representations arising in these models in terms of configurations of projected products of simplices and normal fans of products of simplices. We bound the number of hidden variables, depending on the cardinalities of their state spaces, for which these models can approximate any probability distribution on their visible states to any given accuracy. In addition, we use algebraic methods and coding theory to compute their dimension.

Keywords: restricted Boltzmann machine, naïve Bayes model, representational power, distributed representation, expected dimension

1. Introduction

A restricted Boltzmann machine (RBM) is a probabilistic graphical model with bipartite interactions between an observed set and a hidden set of units (Smolensky, 1986; Freund and Haussler, 1991; Hinton, 2002, 2010). A characterizing property of these models is that the observed units are independent given the states of the hidden units and vice versa. This is a consequence of the bipartiteness of the interaction graph and does not depend on the units' state spaces. Typically RBMs are defined with binary units, but other types of units have also been considered, including continuous, discrete, and mixed type units (Welling et al., 2005; Marks and Movellan, 2001; Salakhutdinov et al., 2007; Dahl et al., 2012; Tran et al., 2011). We study discrete RBMs, also called multinomial or softmax RBMs, which are special types of exponential family harmoniums (Welling et al., 2005). While each unit X_i of a binary RBM has the state space $\{0,1\}$, the state space of each unit X_i of a discrete RBM is a finite set $\mathcal{X}_i = \{0,1,\ldots,r_i-1\}$. Like binary RBMs, discrete RBMs can be trained using contrastive divergence (CD) (Hinton, 1999, 2002; Carreira-Perpiñán and

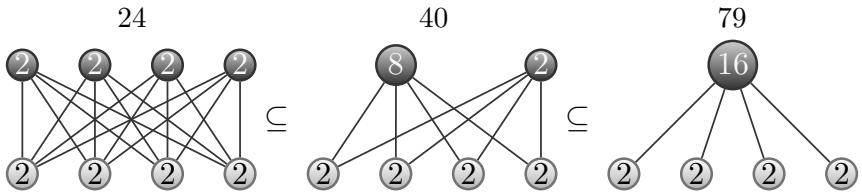


Figure 1: Examples of probability models treated in this paper, in the special case of binary visible variables. The light (dark) nodes represent visible (hidden) variables with the indicated number of states. The total parameter count of each model is indicated at the top. From left to right: a binary RBM; a discrete RBM with one 8-valued and one binary hidden units; and a binary naïve Bayes model with 16 hidden classes.

Hinton, 2005) or expectation-maximization (EM) (Dempster et al., 1977) and can be used to train the parameters of deep systems layer by layer (Hinton et al., 2006; Bengio et al., 2007).

Non-binary visible units are natural because they can directly encode non-binary features. The situation with hidden units is more subtle. States that appear in different hidden units can be activated by the same visible vector, but states that appear in the same hidden unit are mutually exclusive. Non-binary hidden units thus allow one to explicitly represent complex exclusive relationships. For example, a discrete RBM topic model would allow some topics to be mutually exclusive and other topics to be mixed together freely. This provides a better match to the semantics of several learning problems, although the learnability of such representations is mostly open. The practical need to represent mutually exclusive properties is evidenced by the common approach of adding activation sparsity parameters to binary RBM hidden states, which artificially create mutually exclusive non-binary states by penalizing models which have more than a certain percentage of hidden units active.

A discrete RBM is a *product of experts* (Hinton, 1999); each hidden unit represents an expert which is a mixture model of product distributions, or naïve Bayes model. Hence discrete RBMs capture both naïve Bayes models and binary RBMs, and interpolate between non-distributed mixture representations and distributed mixture representations (Bengio, 2009; Montúfar and Morton, 2015). See Figure 1. Naïve Bayes models have been studied across many disciplines. In machine learning they are most commonly used for classification and clustering, but have also been considered for probabilistic modeling (Lowd and Domingos, 2005; Montúfar, 2013). Theoretical work on binary RBM models includes results on universal approximation (Freund and Haussler, 1991; Le Roux and Bengio, 2008; Montúfar and Ay, 2011), dimension and parameter identifiability (Cueto et al., 2010), Bayesian learning coefficients (Aoyagi, 2010), complexity (Long and Servedio, 2010), approximation errors (Montúfar et al., 2011). In this paper we generalize some of these theoretical results to discrete RBMs.

Probability models with more general interactions than strictly bipartite have also been considered, including semi-restricted Boltzmann machines and higher-order interaction Boltzmann machines (Sejnowski, 1986; Memisevic and Hinton, 2010; Osindero and Hinton,

2008; Ranzato et al., 2010). The techniques that we develop in this paper also serve to treat a general class of RBM-like models allowing within-layer interactions, a generalization that will be carried out in a forthcoming work (Montúfar and Morton, 2013).

Section 2 collects basic facts about independence models, naïve Bayes models, and binary RBMs, including an overview on the aforementioned theoretical results. Section 3 defines discrete RBMs formally and describes them as (i) products of mixtures of product distributions (Proposition 6) and (ii) as restricted mixtures of product distributions. Section 4 elaborates on distributed representations and inference functions represented by discrete RBMs (Proposition 9, Lemma 10, and Proposition 11). Section 5 addresses the expressive power of discrete RBMs by describing explicit submodels (Theorem 12) and provides results on their maximal approximation errors and universal approximation properties (Theorem 13). Section 6 treats the dimension of discrete RBM models (Proposition 14 and Theorem 15). Section 7 contains an algebraic-combinatorial discussion of tropical discrete RBM models (Theorem 17) with consequences for their dimension collected in Propositions 20, 21, and 22. Section 8 offers a conclusion.

2. Preliminaries

This section collects basic facts about independence models, naïve Bayes models, and binary RBMs.

2.1 Independence Models

Consider a system of $n < \infty$ random variables $X_1, \dots, X_n$. Assume that X_i takes states x_i in a finite set $\mathcal{X}_i = \{0, 1, \dots, r_i - 1\}$ for all $i \in \{1, \dots, n\} =: [n]$. The state space of this system is $\mathcal{X} := \mathcal{X}_1 \times \cdots \times \mathcal{X}_n$. We write $x_\lambda = (x_i)_{i \in \lambda}$ for a joint state of the variables with index $i \in \lambda$ for any $\lambda \subseteq [n]$, and $x = (x_1, \dots, x_n)$ for a joint state of all variables. We denote by $\Delta(\mathcal{X})$ the set of all probability distributions on $\mathcal{X}$. We write $\langle a, b \rangle$ for the inner product $a^\top b$.

The *independence model* of the variables $X_1, \dots, X_n$ is the set of product distributions $p(x) = \prod_{i \in [n]} p_i(x_i)$ for all $x \in \mathcal{X}$, where p_i is a probability distribution with state space $\mathcal{X}_i$ for all $i \in [n]$. This model is the closure $\overline{\mathcal{E}_\mathcal{X}}$ (in the Euclidean topology) of the exponential family

$$\mathcal{E}_\mathcal{X} := \Big\{ \frac{1}{Z(\theta)} \exp(\langle \theta, A^{(\mathcal{X})} \rangle) \colon \theta \in \mathbb{R}^{d_\mathcal{X}} \Big\},$$

where $A^{(\mathcal{X})} \in \mathbb{R}^{d_\mathcal{X} \times \mathcal{X}}$ is a matrix of sufficient statistics; with rows equal to the indicator functions $\mathbb{1}_\mathcal{X}$ and $\mathbb{1}_{\{x \colon x_i = y_i\}}$ for all $y_i \in \mathcal{X}_i \setminus \{0\}$ for all $i \in [n]$. The partition function $Z(\theta)$ normalizes the distributions. The convex support of $\mathcal{E}_\mathcal{X}$ is the convex hull $Q_\mathcal{X} := \operatorname{conv}(\{A^{(\mathcal{X})}_x\}_{x \in \mathcal{X}})$ of the columns of $A^{(\mathcal{X})}$, which is a Cartesian product of simplices with $Q_\mathcal{X} \cong \Delta(\mathcal{X}_1) \times \cdots \times \Delta(\mathcal{X}_n)$.

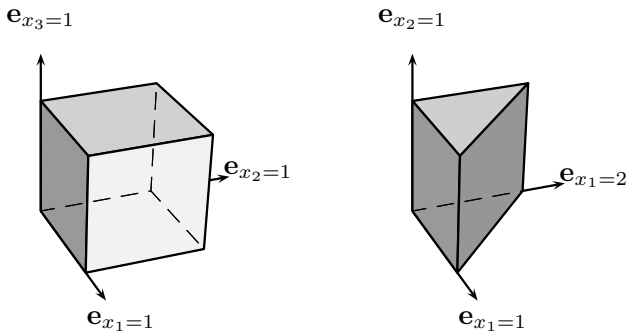


Figure 2: The convex support of the independence model of three binary variables (left) and of a binary-ternary pair of variables (right) discussed in Example 1.

Example 1 The sufficient statistics of the independence models $\mathcal{E}_{\mathcal{X}}$ and $\mathcal{E}_{\mathcal{X}'}$ with state spaces $\mathcal{X} = \{0,1\}^3$ and $\mathcal{X}' = \{0,1,2\}\times\{0,1\}$ are, with rows labeled by indicator functions,

$$A^{(\mathcal{X})} = \begin{array}{c} \begin{bmatrix}1\\1\\1\end{bmatrix}\begin{bmatrix}1\\1\\0\end{bmatrix}\begin{bmatrix}1\\0\\1\end{bmatrix}\begin{bmatrix}1\\0\\0\end{bmatrix}\begin{bmatrix}0\\1\\1\end{bmatrix}\begin{bmatrix}0\\1\\0\end{bmatrix}\begin{bmatrix}0\\0\\1\end{bmatrix}\begin{bmatrix}0\\0\\0\end{bmatrix} \\ \left(\begin{array}{cccccccc} 1&1&1&1&1&1&1&1 \\ \hline 1&1&1&1&0&0&0&0 \\ 1&1&0&0&1&1&0&0 \\ 1&0&1&0&1&0&1&0 \end{array}\right) \end{array} \begin{array}{l} \\ \hline x_3=1 \\ x_2=1 \\ x_1=1 \end{array}$$

$$A^{(\mathcal{X}')} = \begin{array}{c} \begin{bmatrix}1\\2\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix}\begin{bmatrix}1\\0\end{bmatrix}\begin{bmatrix}0\\2\end{bmatrix}\begin{bmatrix}0\\1\end{bmatrix}\begin{bmatrix}0\\0\end{bmatrix} \\ \left(\begin{array}{ccc|ccc} 1&1&1&1&1&1 \\ \hline 1&1&1&0&0&0 \\ \hline 1&0&0&1&0&0 \\ 0&1&0&0&1&0 \end{array}\right) \end{array} \begin{array}{l} \\ \hline x_2=1 \\ \hline x_1=2 \\ x_1=1 \end{array}.$$

In the first case the convex support is a cube and in the second it is a prism. Both convex supports are three-dimensional polytopes, but the prism has fewer vertices and is more similar to a simplex, meaning that its vertex set is affinely more independent than that of the cube. See Figure 2.

2.2 Naïve Bayes Models

Let $k \in \mathbb{N}$. The *k-mixture* of the independence model, or *naïve Bayes model* with k hidden classes, with visible variables $X_1, \ldots, X_n$ is the set of all probability distributions expressible as convex combinations of k points in $\mathcal{E}_{\mathcal{X}}$:

$$\mathcal{M}_{\mathcal{X},k} := \Big\{ \sum_{i\in[k]} \lambda_i p^{(i)} \colon p^{(i)} \in \mathcal{E}_{\mathcal{X}},\ \lambda_i \geq 0,\ \text{for all } i \in [k],\ \text{and } \sum_{i\in[k]} \lambda_i = 1 \Big\}.$$

We write $\mathcal{M}_{n,k}$ for the k-mixture of the independence model of n binary variables. The dimensions of mixtures of binary independence models are known:

Theorem 1 (Catalisano et al. 2011) *The mixtures of binary independence models $\mathcal{M}_{n,k}$ have the dimension expected from counting parameters,* $\min\{nk + (k-1), 2^n - 1\}$*, except for $\mathcal{M}_{4,3}$, which has dimension* 13 *instead of* 14*.*

Let $\mathfrak{A}_{\mathcal{X}}(d)$ denote the maximal cardinality of a subset $\mathcal{X}' \subseteq \mathcal{X}$ of minimum Hamming distance at least d, i.e., the maximal cardinality of a subset $\mathcal{X}' \subseteq \mathcal{X}$ with $d_H(x,y) \geq d$ for

all distinct points $x, y \in \mathcal{X}'$, where $d_H(x,y) := |\{i \in [n] \colon x_i \neq y_i\}|$ denotes the Hamming distance between x and y. The function $\mathfrak{A}_{\mathcal{X}}$ is familiar in coding theory. The k-mixtures of independence models are universal approximators when k is large enough. This can be made precise in terms of $\mathfrak{A}_{\mathcal{X}}(2)$:

Theorem 2 (Montúfar 2013) *The mixture model $\mathcal{M}_{\mathcal{X},k}$ can approximate any probability distribution on $\mathcal{X}$ arbitrarily well if $k \geq |\mathcal{X}|/\max_{i\in[n]} |\mathcal{X}_i|$ and only if $k \geq \mathfrak{A}_{\mathcal{X}}(2)$.*

By results from (Gilbert, 1952; Varshamov, 1957), when q is a power of a prime number and $\mathcal{X} = \{0, 1, \ldots, q-1\}^n$, then $\mathfrak{A}_{\mathcal{X}} = q^{n-1}$. In these cases the previous theorem shows that $\mathcal{M}_{\mathcal{X},k}$ is a universal approximator of distributions on $\mathcal{X}$ if and only if $k \geq q^{n-1}$. In particular, the smallest naïve Bayes model universal approximator of distributions on $\{0,1\}^n$ has $2^{n-1}(n+1)-1$ parameters.

Some of the distributions not representable by a given naïve Bayes model can be characterized in terms of their modes. A state $x \in \mathcal{X}$ is a *mode* of a distribution $p \in \Delta(\mathcal{X})$ if $p(x) > p(y)$ for all y with $d_H(x,y) = 1$ and it is a *strong mode* if $p(x) > \sum_{y\colon d_H(x,y)=1} p(y)$.

Lemma 3 (Montúfar and Morton 2015) *Let $p = \sum_i \lambda_i p^{(i)}$ be a mixture of product distributions. If p has strong modes $\mathcal{C} \subseteq \mathcal{X}$, then there is a mixture component $p^{(i)}$ with mode x for each $x \in \mathcal{C}$.*

2.3 Binary Restricted Boltzmann Machines

The binary RBM model with n visible and m hidden units, denoted $\mathrm{RBM}_{n,m}$, is the set of distributions on $\{0,1\}^n$ of the form

$$p(x) = \frac{1}{Z(W,B,C)} \sum_{h\in\{0,1\}^m} \exp(h^\top W x + B^\top x + C^\top h) \quad \text{for all } x \in \{0,1\}^n, \tag{1}$$

where x denotes states of the visible units, h denotes states of the hidden units, $W = (W_{ji})_{ji} \in \mathbb{R}^{m\times n}$ is a matrix of interaction weights, $B \in \mathbb{R}^n$ and $C \in \mathbb{R}^m$ are vectors of bias weights, and $Z(W,B,C) = \sum_{x\in\{0,1\}^n} \sum_{h\in\{0,1\}^m} \exp(h^\top W x + B^\top x + C^\top h)$ is the normalizing partition function.

It is known that these models have the expected dimension for many choices of n and m:

Theorem 4 (Cueto et al. 2010) *The dimension of the model $\mathrm{RBM}_{n,m}$ is equal to $nm + n + m$ when $m + 1 \leq 2^{n-\lceil \log_2(n+1)\rceil}$ and it is equal to $2^n - 1$ when $m \geq 2^{n-\lfloor \log_2(n+1)\rfloor}$.*

It is also known that with enough hidden units, binary RBMs are universal approximators:

Theorem 5 (Montúfar and Ay 2011) *The model $\mathrm{RBM}_{n,m}$ can approximate any distribution on $\{0,1\}^n$ arbitrarily well whenever $m \geq 2^{n-1} - 1$.*

A previous result by Le Roux and Bengio (2008, Theorem 2) shows that $\mathrm{RBM}_{n,m}$ is a universal approximator whenever $m \geq 2^n + 1$. It is not known whether the bounds from Theorem 5 are always tight, but they show that for any given n, the smallest RBM universal approximator of distributions on $\{0,1\}^n$ has at most $2^{n-1}(n+1)-1$ parameters and hence not more than the smallest naïve Bayes model universal approximator (Theorem 2).

3. Discrete Restricted Boltzmann Machines

Let $\mathcal{X}_i = \{0, 1, \ldots, r_i - 1\}$ for all $i \in [n]$ and $\mathcal{Y}_j = \{0, 1, \ldots, s_j - 1\}$ for all $j \in [m]$. The graphical model with full bipartite interactions $\{\{i, j\} \colon i \in [n], j \in [m]\}$ on $\mathcal{X} \times \mathcal{Y}$ is the exponential family

$$\mathcal{E}_{\mathcal{X},\mathcal{Y}} := \left\{ \frac{1}{Z(\theta)} \exp(\langle \theta, A^{(\mathcal{X},\mathcal{Y})} \rangle) \colon \theta \in \mathbb{R}^{d_{\mathcal{X}} d_{\mathcal{Y}}} \right\}, \tag{2}$$

with sufficient statistics matrix equal to the Kronecker product $A^{(\mathcal{X},\mathcal{Y})} = A^{(\mathcal{X})} \otimes A^{(\mathcal{Y})}$ of the sufficient statistics matrices $A^{(\mathcal{X})}$ and $A^{(\mathcal{Y})}$ of the independence models $\mathcal{E}_{\mathcal{X}}$ and $\mathcal{E}_{\mathcal{Y}}$. The matrix $A^{(\mathcal{X},\mathcal{Y})}$ has $d_{\mathcal{X}} d_{\mathcal{Y}} = \left(\sum_{i \in [n]} (|\mathcal{X}_i| - 1) + 1\right) \left(\sum_{j \in [m]} (|\mathcal{Y}_i| - 1) + 1\right)$ linearly independent rows and $|\mathcal{X} \times \mathcal{Y}|$ columns, each column corresponding to a joint state (x, y) of all variables. Disregarding the entry of θ that is multiplied with the constant row of $A^{(\mathcal{X},\mathcal{Y})}$, which cancels out with the normalization function $Z(\theta)$, this parameterization of $\mathcal{E}_{\mathcal{X},\mathcal{Y}}$ is one-to-one. In particular, this model has dimension $\dim(\mathcal{E}_{\mathcal{X},\mathcal{Y}}) = d_{\mathcal{X}} d_{\mathcal{Y}} - 1$.

The discrete RBM model $\mathrm{RBM}_{\mathcal{X},\mathcal{Y}}$ is the following set of marginal distributions:

$$\mathrm{RBM}_{\mathcal{X},\mathcal{Y}} := \left\{ q(x) = \sum_{y \in \mathcal{Y}} p(x, y) \text{ for all } x \in \mathcal{X} \colon p \in \mathcal{E}_{\mathcal{X},\mathcal{Y}} \right\}.$$

In the case of one single hidden unit, this model is the naïve Bayes model on $\mathcal{X}$ with $|\mathcal{Y}_1|$ hidden classes. When all units are binary, $\mathcal{X} = \{0, 1\}^n$ and $\mathcal{Y} = \{0, 1\}^m$, this model is $\mathrm{RBM}_{n,m}$. Note that the exponent in Equation 1 can be written as $(h^\top W x + B^\top x + C^\top h) = \langle \theta, A^{(\mathcal{X},\mathcal{Y})}_{(x,h)} \rangle$, taking for θ the column-by-column vectorization of the matrix $\left(\begin{smallmatrix} 0 & B^\top \\ C & W \end{smallmatrix}\right)$.

3.1 Conditional Distributions

The conditional distributions of discrete RBMs can be described in the following way. Consider a vector $\theta \in \mathbb{R}^{d_{\mathcal{X}} d_{\mathcal{Y}}}$ parameterizing $\mathcal{E}_{\mathcal{X},\mathcal{Y}}$, and the matrix $\Theta \in \mathbb{R}^{d_{\mathcal{Y}} \times d_{\mathcal{X}}}$ with column-by-column vectorization equal to θ. A lemma by Roth (1934) shows that $\theta^\top (A^{(\mathcal{X})} \otimes A^{(\mathcal{Y})})_{(x,y)} = (A^{(\mathcal{X})}_x)^\top \Theta^\top A^{(\mathcal{Y})}_y$ for all $x \in \mathcal{X}$, $y \in \mathcal{Y}$, and hence

$$\left\langle \theta, A^{(\mathcal{X},\mathcal{Y})}_{(x,y)} \right\rangle = \left\langle \Theta A^{(\mathcal{X})}_x, A^{(\mathcal{Y})}_y \right\rangle = \left\langle \Theta^\top A^{(\mathcal{Y})}_y, A^{(\mathcal{X})}_x \right\rangle \quad \forall x \in \mathcal{X}, y \in \mathcal{Y}. \tag{3}$$

The inner product in Equation 3 describes following probability distributions:

$$\begin{aligned} p_\theta(\cdot, \cdot) &= \frac{1}{Z(\theta)} \exp\left(\left\langle \theta, A^{(\mathcal{X},\mathcal{Y})} \right\rangle\right), \\ p_\theta(\cdot | x) &= \frac{1}{Z\left(\Theta A^{(\mathcal{X})}_x\right)} \exp\left(\left\langle \Theta A^{(\mathcal{X})}_x, A^{(\mathcal{Y})} \right\rangle\right), \text{ and} \\ p_\theta(\cdot | y) &= \frac{1}{Z\left(\Theta^\top A^{(\mathcal{Y})}_y\right)} \exp\left(\left\langle \Theta^\top A^{(\mathcal{Y})}_y, A^{(\mathcal{X})} \right\rangle\right). \end{aligned}$$

Geometrically, $\Theta A^{(\mathcal{X})}$ is a linear projection of the columns of the sufficient statistics matrix $A^{(\mathcal{X})}$ into the parameter space of $\mathcal{E}_{\mathcal{Y}}$, and similarly, $\Theta^\top A^{(\mathcal{Y})}$ is a linear projection of the columns of $A^{(\mathcal{Y})}$ into the parameter space of $\mathcal{E}_{\mathcal{X}}$.

3.2 Polynomial Parameterization

Discrete RBMs can be parameterized not only in the exponential way discussed above, but also by simple polynomials. The exponential family $\mathcal{E}_{\mathcal{X},\mathcal{Y}}$ can be parameterized by square free monomials:

$$p(v,h) = \frac{1}{Z} \prod_{\substack{\{j,i\} \in [m] \times [n], \\ (y'_j, x'_i) \in \mathcal{Y}_j \times \mathcal{X}_i}} \left(\gamma_{\{j,i\},(y'_j,x'_i)}\right)^{\delta_{y'_j}(h_j)\delta_{x'_i}(v_i)} \quad \text{for all } (v,h) \in \mathcal{Y} \times \mathcal{X},$$

where $\gamma_{\{j,i\},(y'_j,x'_i)}$ are positive reals. The probability distributions in $\mathrm{RBM}_{\mathcal{X},\mathcal{Y}}$ can be written as

$$p(v) = \frac{1}{Z} \prod_{j \in [m]} \Big(\sum_{h_j \in \mathcal{Y}_j} \gamma_{\{j,1\},(h_j,v_1)} \cdots \gamma_{\{j,n\},(h_j,v_n)} \Big) \quad \text{for all } v \in \mathcal{X}. \tag{4}$$

The parameters $\gamma_{\{j,i\},(y'_j,x'_i)}$ correspond to $\exp(\theta_{\{j,i\},(y'_j,x'_i)})$ in the parameterization given in Equation 2.

3.3 Products of Mixtures and Mixtures of Products

In the following we describe discrete RBMs from two complementary perspectives: (i) as products of experts, where each expert is a mixture of products, and (ii) as restricted mixtures of product distributions. The renormalized entry-wise (Hadamard) product of two probability distributions p and q on $\mathcal{X}$ is defined as $p \circ q := (p(x)q(x))_{x \in \mathcal{X}} / \sum_{y \in \mathcal{X}} p(y)q(y)$. Here we assume that p and q have overlapping supports, such that the definition makes sense.

Proposition 6 *The model* $\mathrm{RBM}_{\mathcal{X},\mathcal{Y}}$ *is a Hadamard product of mixtures of product distributions:*

$$\mathrm{RBM}_{\mathcal{X},\mathcal{Y}} = \mathcal{M}_{\mathcal{X},|\mathcal{Y}_1|} \circ \cdots \circ \mathcal{M}_{\mathcal{X},|\mathcal{Y}_m|}\,.$$

Proof The statement can be seen directly by considering the parameterization from Equation 4. To make this explicit, one can use a *homogeneous* version of the matrix $A^{(\mathcal{X},\mathcal{Y})}$ which we denote by A and which defines the same model. Each row of A is indexed by an edge $\{i,j\}$ of the bipartite graph and a joint state (x_i, h_j) of the visible and hidden units connected by this edge. Such a row has a one in any column when these states agree with the global state, and zero otherwise. For any $j \in [m]$ let $A_{j,:}$ denote the matrix containing the rows of A with indices $(\{i,j\},(x_i,h_j))$ for all $x_i \in \mathcal{X}_i$ for all $i \in [n]$ for all $h_j \in \mathcal{Y}_j$, and let $A(x,h)$ denote the (x,h)-column of A. We have

$$\begin{aligned}
p(x) =& \frac{1}{Z} \sum_h \exp(\langle \theta, A(x,h) \rangle) \\
=& \frac{1}{Z} \sum_h \exp(\langle \theta_{1,:}, A_{1,:}(x,h) \rangle) \exp(\langle \theta_{2,:}, A_{2,:}(x,h) \rangle) \cdots \exp(\langle \theta_{m,:}, A_{m,:}(x,h) \rangle) \\
=& \frac{1}{Z} \Big(\sum_{h_1} \exp(\langle \theta_{1,:}, A_{1,:}(x,h_1) \rangle) \Big) \cdots \Big(\sum_{h_m} \exp(\langle \theta_{m,:}, A_{m,:}(x,h_m) \rangle) \Big) \\
=& \frac{1}{Z} (Z_1 p^{(1)}(x)) \cdots (Z_m p^{(m)}(x)) = \frac{1}{Z'} p^{(1)}(x) \cdots p^{(m)}(x),
\end{aligned}$$

where $p^{(j)} \in \mathcal{M}_{\mathcal{X},|\mathcal{Y}_j|}$ and $Z_j = \sum_{x\in\mathcal{X}} \sum_{h_j\in\mathcal{Y}_j} \exp(\langle \theta_{j,:}, A_{j,:}(x,h_j)\rangle)$ for all $j \in [m]$. Since the vectors $\theta_{j,:}$ can be chosen arbitrarily, the factors $p^{(j)}$ can be made arbitrary within $\mathcal{M}_{\mathcal{X},|\mathcal{Y}_j|}$. ■

Of course, every distribution in $\mathrm{RBM}_{\mathcal{X},\mathcal{Y}}$ is a mixture distribution $p(x) = \sum_{h\in\mathcal{Y}} p(x|h)q(h)$. The mixture weights are given by the marginals $q(h)$ on $\mathcal{Y}$ of distributions from $\mathcal{E}_{\mathcal{X},\mathcal{Y}}$, and the mixture components can be described as follows.

Proposition 7 *The set of conditional distributions $p(\cdot|h)$, $h \in \mathcal{Y}$ of a distribution in $\mathcal{E}_{\mathcal{X},\mathcal{Y}}$ is the set of product distributions in $\mathcal{E}_{\mathcal{X}}$ with parameters $\theta_h = \Theta^\top A^{(\mathcal{Y})}_h$, $h \in \mathcal{Y}$ equal to a linear projection of the vertices $\{A^{(\mathcal{Y})}_h : h \in \mathcal{Y}\}$ of the Cartesian product of simplices $Q_{\mathcal{Y}} \cong \Delta(\mathcal{Y}_1) \times \cdots \times \Delta(\mathcal{Y}_m)$.*

Proof This is by Equation 3. ■

4. Products of Simplices and Their Normal Fans

Binary RBMs have been analyzed by considering each of the m hidden units as defining a hyperplane H_j slicing the n-cube into two regions. To generalize the results provided by this analysis, in this section we replace the n-cube with a general product of simplices $Q_{\mathcal{X}}$, and replace the two regions defined by the hyperplane H_j by the $|\mathcal{Y}_j|$ regions defined by the maximal cones of the normal fan of the simplex $\Delta(\mathcal{Y}_j)$.

4.1 Subdivisions of Independence Models

The *normal cone* of a polytope $Q \subset \mathbb{R}^d$ at a point $x \in Q$ is the set of all vectors $v \in \mathbb{R}^d$ with $\langle v, (x-y)\rangle \geq 0$ for all $y \in Q$. We denote by R_x the normal cone of the product of simplices $Q_{\mathcal{X}} = \mathrm{conv}\{A^{(\mathcal{X})}_x\}_{x\in\mathcal{X}}$ at the vertex $A^{(\mathcal{X})}_x$. The normal fan $\mathcal{F}_{\mathcal{X}}$ is the set of all normal cones of $Q_{\mathcal{X}}$. The product distributions $p_\theta = \frac{1}{Z(\theta)} \exp(\langle \theta, A^{(\mathcal{X})}\rangle) \in \mathcal{E}_{\mathcal{X}}$ strictly maximized at $x \in \mathcal{X}$, with $p_\theta(x) > p_\theta(y)$ for all $y \in \mathcal{X} \setminus \{x\}$, are those with parameter vector θ in the relative interior of R_x. Hence the normal fan $\mathcal{F}_{\mathcal{X}}$ partitions the parameter space of the independence model into regions of distributions with maxima at different inputs.

4.2 Inference Functions and Slicings

For any choice of parameters of the model $\mathrm{RBM}_{\mathcal{X},\mathcal{Y}}$, there is an *inference function* $\pi : \mathcal{X} \to \mathcal{Y}$, (or more generally $\pi : \mathcal{X} \to 2^{\mathcal{Y}}$), which computes the most likely hidden state given a visible state. These functions are not necessarily injective nor surjective. For a visible state x, the conditional distribution on the hidden states is a product distribution $p(y|X = x) = \frac{1}{Z} \exp(\langle \Theta A^{(\mathcal{X})}_x, A^{(\mathcal{Y})}_y\rangle)$ which is maximized at the state y for which $\Theta A^{(\mathcal{X})}_x \in R_y$. The preimages of the cones R_y by the map Θ partition the input space $\mathbb{R}^{d_{\mathcal{X}}}$ and are called *inference regions*. See Figure 3 and Example 2.

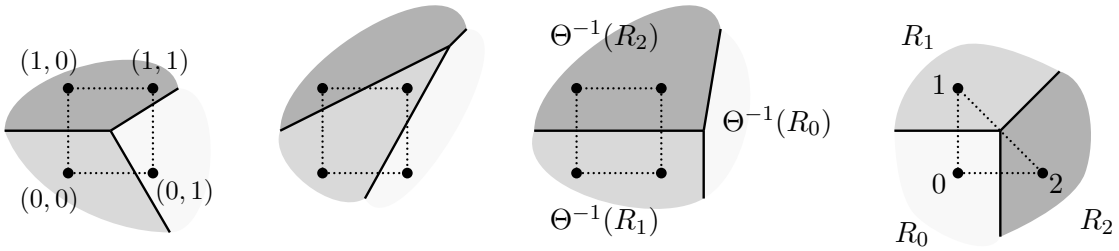


Figure 3: Three slicings of a square by the normal fan of a triangle with maximal cones R_0, R_1, and R_2, corresponding to three possible inference functions of $\mathrm{RBM}_{\{0,1\}^2,\{0,1,2\}}$.

Definition 8 A $\mathcal{Y}$-*slicing* of a finite set $\mathcal{Z} \subset \mathbb{R}^{d_\mathcal{X}}$ is a partition of $\mathcal{Z}$ into the preimages of the cones R_y, $y \in \mathcal{Y}$ by a linear map $\Theta\colon \mathbb{R}^{d_\mathcal{X}} \to \mathbb{R}^{d_\mathcal{Y}}$. We assume that Θ is generic, such that it maps each element of $\mathcal{Z}$ into the interior of some R_y.

For example, when $\mathcal{Y} = \{0, 1\}$, the fan $\mathcal{F}_\mathcal{Y}$ consists of a hyperplane and the two closed half-spaces defined by that hyperplane. A $\mathcal{Y}$-slicing is in this case a standard slicing by a hyperplane.

Example 2 Let $\mathcal{X} = \{0, 1, 2\} \times \{0, 1\}$ and $\mathcal{Y} = \{0, 1\}^4$. The maximal cones R_y, $y \in \mathcal{Y}$ of the normal fan of the 4-cube with vertices $\{0, 1\}^4$ are the closed orthants of $\mathbb{R}^4$. The 6 vertices $\{A_x^{(\mathcal{X})}\colon x \in \mathcal{X}\}$ of the prism $\Delta(\{0, 1, 2\}) \times \Delta(\{0, 1\})$ can be mapped into 6 distinct orthants of $\mathbb{R}^4$, each orthant with an even number of positive coordinates:

$$\underset{\Theta}{\begin{pmatrix} 3 & -2 & -2 & -2 \\ 1 & 2 & -2 & -2 \\ 1 & -2 & -2 & 2 \\ 1 & -2 & 2 & -2 \end{pmatrix}} \underset{A^{(\mathcal{X})}}{\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{pmatrix}} = \begin{pmatrix} -1 & -1 & 1 & 1 & 1 & 3 \\ 1 & 1 & 3 & -1 & -1 & 1 \\ -3 & 1 & -1 & -1 & 3 & 1 \\ 1 & -3 & -1 & 3 & -1 & 1 \end{pmatrix}.$$

Even in the case of one single hidden unit the slicings can be complex, but the following simple type of slicing is always available.

Proposition 9 *Any slicing by $k-1$ parallel hyperplanes is a $\{1, 2, \ldots, k\}$-slicing.*

Proof We show that there is a line $\mathcal{L} = \{\lambda r - b\colon \lambda \in \mathbb{R}\}$, $r, b \in \mathbb{R}^k$ intersecting all cells of $\mathcal{F}_\mathcal{Y}$, $\mathcal{Y} = \{1, \ldots, k\}$. We need to show that there is a choice of r and b such that for every $y \in \mathcal{Y}$ the set $I_y \subseteq \mathbb{R}$ of all λ with $\langle \lambda r - b, (\mathbf{e}_y - \mathbf{e}_z)\rangle > 0$ for all $z \in \mathcal{Y} \setminus \{y\}$ has a non-empty interior. Now, I_y is the set of λ with

$$\lambda(r_y - r_z) > b_y - b_z \quad \text{for all } z \neq y.$$

Choosing $b_1 < \cdots < b_k$ and $r_y = f(b_y)$, where f is a strictly increasing and strictly concave function, we get $I_1 = (-\infty, \frac{b_2 - b_1}{r_2 - r_1})$, $I_y = (\frac{b_y - b_{y-1}}{r_y - r_{y-1}}, \frac{b_{y+1} - b_y}{r_{y+1} - r_y})$ for $y = 2, 3, \ldots, k-1$, and

$I_k = (\frac{b_k - b_{k-1}}{r_k - r_{k-1}}, \infty)$. The lengths $\infty, l_2, \ldots, l_{k-1}, \infty$ of the intervals $I_1, \ldots, I_k$ can be adjusted arbitrarily by choosing suitable differences $r_{j+1} - r_j$ for all $j = 1, \ldots, k-1$. ■

4.3 Strong Modes

Recall the definition of strong modes given in page 657.

Lemma 10 *Let $\mathcal{C} \subseteq \mathcal{X}$ be a set of arrays which are pairwise different in at least two entries (a code of minimum distance two).*

- *If $\mathrm{RBM}_{\mathcal{X},\mathcal{Y}}$ contains a probability distribution with strong modes $\mathcal{C}$, then there is a linear map Θ of $\{A^{(\mathcal{Y})}_y : y \in \mathcal{Y}\}$ into the $\mathcal{C}$-cells of $\mathcal{F}_{\mathcal{X}}$ (the cones R_x above the code words $x \in \mathcal{C}$) sending at least one vertex into each cell.*
- *If there is a linear map Θ of $\{A^{(\mathcal{Y})}_y : y \in \mathcal{Y}\}$ into the $\mathcal{C}$-cells of $\mathcal{F}_{\mathcal{X}}$, with*

$$\max_x \{\langle \Theta^\top A^{(\mathcal{Y})}_y, A^{(\mathcal{X})}_x \rangle\} = c$$

for all $y \in \mathcal{Y}$, then $\mathrm{RBM}_{\mathcal{X},\mathcal{Y}}$ contains a probability distribution with strong modes $\mathcal{C}$.

Proof This is by Proposition 7 and Lemma 3. ■

A simple consequence of the previous lemma is that if the model $\mathrm{RBM}_{\mathcal{X},\mathcal{Y}}$ is a universal approximator of distributions on $\mathcal{X}$, then necessarily the number of hidden states is at least as large as the maximum code of visible states of minimum distance two, $|\mathcal{Y}| \geq \mathfrak{A}_{\mathcal{X}}(2)$. Hence discrete RBMs may not be universal approximators even when their parameter count surpasses the dimension of the ambient probability simplex.

Example 3 Let $\mathcal{X} = \{0,1,2\}^n$ and $\mathcal{Y} = \{0,1,\ldots,4\}^m$. In this case $\mathfrak{A}_{\mathcal{X}}(2) = 3^{n-1}$. If $\mathrm{RBM}_{\mathcal{X},\mathcal{Y}}$ is a universal approximator with $n = 3$ and $n = 4$, then $m \geq 2$ and $m \geq 3$, respectively, although the smallest m for which $\mathrm{RBM}_{\mathcal{X},\mathcal{Y}}$ has $3^n - 1$ parameters is $m = 1$ and $m = 2$, respectively.

Using Lemma 10 and the analysis by Montúfar and Morton (2015) gives the following.

Proposition 11 *If $4\lceil m/3 \rceil \leq n$, then $\mathrm{RBM}_{\mathcal{X},\mathcal{Y}}$ contains distributions with 2^m strong modes.*

5. Representational Power and Approximation Errors

In this section we describe submodels of discrete RBMs and use them to provide bounds on the model approximation errors depending on the number of units and their state spaces. Universal approximation results follow as special cases with vanishing approximation error.

Theorem 12 *The model $\mathrm{RBM}_{\mathcal{X},\mathcal{Y}}$ can approximate the following arbitrarily well:*

- *Any mixture of* $d_{\mathcal{Y}} = 1 + \sum_{j=1}^{m}(|\mathcal{Y}_j| - 1)$ *product distributions with disjoint supports.*
- *When* $d_{\mathcal{Y}} \geq (\prod_{i\in[k]} |\mathcal{X}_i|)/\max_{j\in[k]} |\mathcal{X}_j|$ *for some* $k \leq n$*, any distribution from the model* $\mathcal{P}$ *of distributions with constant value on each block* $\{x_1\} \times \cdots \times \{x_k\} \times \mathcal{X}_{k+1} \times \cdots \times \mathcal{X}_n$ *for all* $x_i \in \mathcal{X}_i$*, for all* $i \in [k]$*.*
- *Any probability distribution with support contained in the union of* $d_{\mathcal{Y}}$ *sets of the form* $\{x_1\} \times \cdots \times \{x_{k-1}\} \times \mathcal{X}_k \times \{x_{k+1}\} \times \cdots \times \{x_n\}$*.*

Proof By Proposition 6 the model $\mathrm{RBM}_{\mathcal{X},\mathcal{Y}}$ contains any Hadamard product $p^{(1)} \circ \cdots \circ p^{(m)}$ with mixtures of products as factors, $p^{(j)} \in \mathcal{M}_{\mathcal{X},|\mathcal{Y}_j|}$ for all $j \in [m]$. In particular, it contains $p = p^{(0)} \circ (\mathbb{1} + \tilde{\lambda}_1 \tilde{p}^{(1)}) \circ \cdots \circ (\mathbb{1} + \tilde{\lambda}_m \tilde{p}^{(m)})$, where $p^{(0)} \in \mathcal{E}_{\mathcal{X}}$, $\tilde{p}^{(j)} \in \mathcal{M}_{\mathcal{X},|\mathcal{Y}_j|-1}$, and $\tilde{\lambda}_j \in \mathbb{R}_+$. Choosing the factors $\tilde{p}^{(j)}$ with pairwise disjoint supports shows that $p = \sum_{j=0}^{m} \lambda_j p^{(j)}$, whereby $p^{(0)}$ can be any product distribution and $p^{(j)}$ can be any distribution from $\mathcal{M}_{\mathcal{X},|\mathcal{Y}_j|-1}$ for all $j \in [m]$, as long as $\operatorname{supp}(p^{(j)}) \cap \operatorname{supp}(p^{(j')})$ for all $j \neq j'$. This proves the first item.

For the second item: Any point in the set $\mathcal{P}$ is a mixture of uniform distributions supported on the disjoint blocks $\{x_1\} \times \cdots \times \{x_k\} \times \mathcal{X}_{k+1} \times \cdots \times \mathcal{X}_n$ for all $(x_1, \ldots, x_k) \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_k$. Each of these uniform distributions is a product distribution, since it factorizes as $p_{x_1,\ldots,x_k} = \prod_{i\in[k]} \delta_{x_i} \prod_{i\in[n]\setminus[k]} u_i$, where u_i denotes the uniform distribution on $\mathcal{X}_i$. For any $j \in [k]$ any mixture $\sum_{x_j\in\mathcal{X}_j} \lambda_{x_j} p_{x_1,\ldots,x_k}$ is also a product distribution, since it factorizes as

$$\Big(\sum_{x_j\in\mathcal{X}_j} \lambda_{x_j}\delta_{x_j}\Big) \prod_{i\in[k]\setminus\{j\}} \delta_{x_i} \prod_{i\in[n]\setminus[k]} u_i.$$

Hence any distribution from the set $\mathcal{P}$ is a mixture of $(\prod_{i\in[k]} |\mathcal{X}_i|)/\max_{j\in[k]} |\mathcal{X}_j|$ product distributions with disjoint supports. The claim now follows from the first item.

For the third item: The model $\overline{\mathcal{E}_{\mathcal{X}}}$ contains any distribution with support of the form $\{x_1\} \times \cdots \times \{x_{k-1}\} \times \mathcal{X}_k \times \{x_{k+1}\} \times \cdots \times \{x_n\}$. Hence, by the first item, the RBM model can approximate any distribution arbitrarily well whose support can be covered by $d_{\mathcal{Y}}$ sets of that form. ■

We now analyze the RBM model approximation errors. Let p and q be two probability distributions on $\mathcal{X}$. The Kullback-Leibler divergence from p to q is defined as $D(p\|q) := \sum_{x\in\mathcal{X}} p(x) \log \frac{p(x)}{q(x)}$ when $\operatorname{supp}(p) \subseteq \operatorname{supp}(q)$ and $D(p\|q) := \infty$ otherwise. The divergence from p to a model $\mathcal{M} \subseteq \Delta(\mathcal{X})$ is defined as $D(p\|\mathcal{M}) := \inf_{q\in\mathcal{M}} D(p\|q)$ and the maximal approximation error of $\mathcal{M}$ is $\sup_{p\in\Delta(\mathcal{X})} D(p\|\mathcal{M})$.

It is known that the maximal approximation error of the independence model $\mathcal{E}_{\mathcal{X}}$ satisfies $\sup_{p\in\Delta(\mathcal{X})} D(p\|\mathcal{E}_{\mathcal{X}}) \leq |\mathcal{X}|/\max_{i\in[n]} |\mathcal{X}_i|$, with equality when all units have the same number of states (Ay and Knauf, 2006, Corollary 4.10).

Theorem 13 *If* $\prod_{i\in[n]\setminus\Lambda} |\mathcal{X}_i| \leq 1 + \sum_{j\in[m]}(|\mathcal{Y}_j| - 1) = d_{\mathcal{Y}}$ *for some* $\Lambda \subseteq [n]$*, then the Kullback-Leibler divergence from any distribution* p *on* $\mathcal{X}$ *to the model* $\mathrm{RBM}_{\mathcal{X},\mathcal{Y}}$ *is bounded by*

$$D(p\|\,\mathrm{RBM}_{\mathcal{X},\mathcal{Y}}) \leq \log \frac{\prod_{i\in\Lambda} |\mathcal{X}_i|}{\max_{i\in\Lambda} |\mathcal{X}_i|}.$$

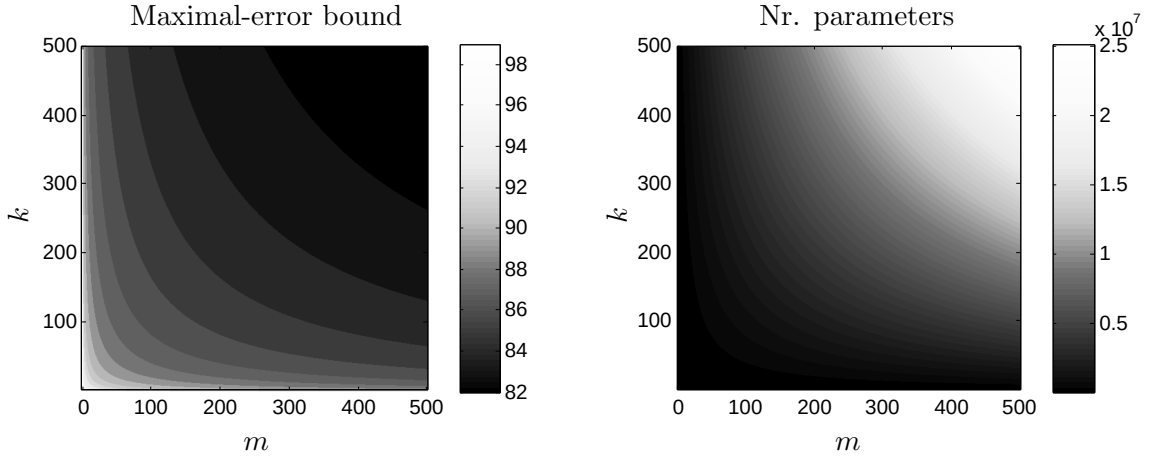


Figure 4: Illustration of Theorem 13. The left panel shows a heat map of the upper bound on the Kullback-Leibler approximation errors of discrete RBMs with 100 visible binary units and the right panel shows a map of the total number of model parameters, both depending on the number of hidden units m and their possible states $k = |\mathcal{Y}_j|$ for all $j \in [m]$.

In particular, the model $\mathrm{RBM}_{\mathcal{X},\mathcal{Y}}$ *is a universal approximator whenever* $d_{\mathcal{Y}} \geq |\mathcal{X}|/\max_{i\in[n]} |\mathcal{X}_i|$.

Proof The submodel $\mathcal{P}$ of $\mathrm{RBM}_{\mathcal{X},\mathcal{Y}}$ described in the second item of Theorem 12 is a *partition model*. The maximal divergence from such a model is equal to the logarithm of the cardinality of the largest block with constant values (Matúš and Ay, 2003). Thus $\max_p D(p\|\,\mathrm{RBM}_{\mathcal{X},\mathcal{Y}}) \leq \max_p D(p\|\mathcal{P}) = \log\big((\prod_{i\in\Lambda} |\mathcal{X}_i|)/\max_{i\in\Lambda} |\mathcal{X}_i|\big)$, as was claimed. ■

Theorem 13 shows that, on a large scale, the maximal model approximation error of $\mathrm{RBM}_{\mathcal{X},\mathcal{Y}}$ is smaller than that of the independence model $\mathcal{E}_{\mathcal{X}}$ by at least $\log(1+\sum_{j\in[m]}(|\mathcal{Y}_j|-1))$, or vanishes. The theorem is illustrated in Figure 4. The line $k = 2$ shows bounds on the approximation error of binary RBMs with m hidden units, previously treated in (Montúfar et al., 2011, Theorem 5.1), and the line $m = 1$ shows bounds for naïve Bayes models with k hidden classes.

6. Dimension

In this section we study the dimension of the model $\mathrm{RBM}_{\mathcal{X},\mathcal{Y}}$. One reason RBMs are attractive is that they have a large learning capacity, e.g. may be built with millions of parameters. Dimension calculations show whether those parameters are wasted, or translate into higher-dimensional spaces of representable distributions. Our analysis builds on previous work by Cueto, Morton, and Sturmfels (2010), where binary RBMs are treated. The idea is to bound the dimension from below by the dimension of a related max-plus model, called the tropical RBM model (Pachter and Sturmfels, 2004), and from above by the dimension expected from counting parameters.

The dimension of a discrete RBM model can be bounded from above not only by its expected dimension, but also by a function of the dimension of its Hadamard factors:

Proposition 14 *The dimension of* $\mathrm{RBM}_{\mathcal{X},\mathcal{Y}}$ *is bounded as*

$$\dim(\mathrm{RBM}_{\mathcal{X},\mathcal{Y}}) \leq \dim(\mathcal{M}_{\mathcal{X},|\mathcal{Y}_i|}) + \sum_{j\in[m]\setminus\{i\}} \dim(\mathcal{M}_{\mathcal{X},|\mathcal{Y}_j|-1}) + (m-1) \quad \textit{for all } i \in [m]. \tag{5}$$

Proof Let u denote the uniform distribution. Note that $\mathcal{E}_{\mathcal{X}} \circ \mathcal{E}_{\mathcal{X}} = \mathcal{E}_{\mathcal{X}}$ and also $\mathcal{E}_{\mathcal{X}} \circ \mathcal{M}_{\mathcal{X},k} = \mathcal{M}_{\mathcal{X},k}$. This observation, together with Proposition 6, shows that the RBM model can be factorized as

$$\mathrm{RBM}_{\mathcal{X},\mathcal{Y}} = (\mathcal{M}_{\mathcal{X},|\mathcal{Y}_1|}) \circ (\lambda_1 u + (1-\lambda_1)\mathcal{M}_{\mathcal{X},|\mathcal{Y}_1|}) \circ \cdots \circ (\lambda_m u + (1-\lambda_m)\mathcal{M}_{\mathcal{X},|\mathcal{Y}_m|-1}),$$

from which the claim follows. ■

By the previous proposition, the model $\mathrm{RBM}_{\mathcal{X},\mathcal{Y}}$ can have the expected dimension only if (i) the right hand side of Equation 5 equals $|\mathcal{X}|-1$, or (ii) each mixture model $\mathcal{M}_{\mathcal{X},k}$ has the expected dimension for all $k \leq \max_{j\in[m]} |\mathcal{Y}_j|$. Sometimes none of both conditions is satisfied and the models 'waste' parameters:

Example 4 The k-mixture of the independence model on $\mathcal{X}_1 \times \mathcal{X}_2$ is a subset of the set of $|\mathcal{X}_1| \times |\mathcal{X}_2|$ matrices with non-negative entries and rank at most k. It is known that the set of $M \times N$ matrices of rank at most k has dimension $k(M+N-k)$ for all $1 \leq k < \min\{M,N\}$. Hence the model $\mathcal{M}_{\mathcal{X}_1\times\mathcal{X}_2,k}$ has dimension smaller than its parameter count whenever $1 < k < \min\{|\mathcal{X}_1|,|\mathcal{X}_2|\}$. By Proposition 14 if $(\sum_{j\in[m]}(|\mathcal{Y}_j|-1)+1)(|\mathcal{X}_1|+|\mathcal{X}_2|-1) \leq |\mathcal{X}_1 \times \mathcal{X}_2|$ and $1 < |\mathcal{Y}_j| \leq \min\{|\mathcal{X}_1|,|\mathcal{X}_2|\}$ for some $j \in [m]$, then $\mathrm{RBM}_{\mathcal{X}_1\times\mathcal{X}_2,\mathcal{Y}}$ does not have the expected dimension.

The next theorem indicates choices of $\mathcal{X}$ and $\mathcal{Y}$ for which the model $\mathrm{RBM}_{\mathcal{X},\mathcal{Y}}$ has the expected dimension. Given a sufficient statistics matrix $A^{(\mathcal{X})}$, we say that a set $\mathcal{Z} \subseteq \mathcal{X}$ has full rank when the matrix with columns $\{A^{(\mathcal{X})}_x : x \in \mathcal{Z}\}$ has full rank.

Theorem 15 *When* $\mathcal{X}$ *contains* m *disjoint Hamming balls of radii* $2(|\mathcal{Y}_j|-1)-1$, $j \in [m]$ *and the subset of* $\mathcal{X}$ *not intersected by these balls has full rank, then the model* $\mathrm{RBM}_{\mathcal{X},\mathcal{Y}}$ *has dimension equal to the number of model parameters,*

$$\dim(\mathrm{RBM}_{\mathcal{X},\mathcal{Y}}) = (1 + \sum_{i\in[n]}(|\mathcal{X}_i|-1))(1 + \sum_{j\in[m]}(|\mathcal{Y}_j|-1)) - 1.$$

On the other hand, if m *Hamming balls of radius one cover* $\mathcal{X}$*, then*

$$\dim(\mathrm{RBM}_{\mathcal{X},\mathcal{Y}}) = |\mathcal{X}| - 1.$$

In order to prove this theorem we will need two main tools: slicings by normal fans of simplices, described in Section 4, and the tropical RBM model, described in Section 7. The theorem will follow from the analysis contained in Section 7.

7. Tropical Model

Definition 16 The tropical model $\mathrm{RBM}^{\mathrm{tropical}}_{\mathcal{X},\mathcal{Y}}$ is the image of the tropical morphism

$$\mathbb{R}^{d_{\mathcal{X}} d_{\mathcal{Y}}} \ni \theta \;\; \mapsto \;\; \Phi(v;\theta) = \max\{\langle \theta, A^{(\mathcal{X},\mathcal{Y})}_{(v,h)} \rangle \colon h \in \mathcal{Y}\} \quad \text{for all } v \in \mathcal{X},$$

which evaluates $\log(\frac{1}{Z(\theta)} \sum_{h\in\mathcal{Y}} \exp(\langle \theta, A^{(\mathcal{X},\mathcal{Y})}_{(v,h)} \rangle))$ for all $v \in \mathcal{X}$ for each θ within the max-plus algebra (addition becomes $a + b = \max\{a, b\}$) up to additive constants independent of v (i.e., disregarding the normalization factor $Z(\theta)$).

The idea behind this definition is that $\log(\exp(a) + \exp(b)) \approx \max\{a, b\}$ when a and b have different order of magnitude. The tropical model captures important properties of the original model. Of particular interest is following consequence of the Bieri-Groves theorem (Draisma, 2008), which gives us a tool to estimate the dimension of $\mathrm{RBM}_{\mathcal{X},\mathcal{Y}}$:

$$\dim(\mathrm{RBM}^{\mathrm{tropical}}_{\mathcal{X},\mathcal{Y}}) \leq \dim(\mathrm{RBM}_{\mathcal{X},\mathcal{Y}}) \leq \min\{\dim(\mathcal{E}_{\mathcal{X},\mathcal{Y}}), |\mathcal{X}| - 1\}.$$

The following Theorem 17 describes the regions of linearity of the map Φ. Each of these regions corresponds to a collection of $\mathcal{Y}_j$-slicings (see Definition 8) of the set $\{A^{(\mathcal{X})}_x \colon x \in \mathcal{X}\}$ for all $j \in [m]$. This result allows us to express the dimension of $\mathrm{RBM}^{\mathrm{tropical}}_{\mathcal{X},\mathcal{Y}}$ as the maximum rank of a class of matrices defined by collections of slicings.

For each $j \in [m]$ let $C_j = \{C_{j,1}, \ldots, C_{j,|\mathcal{Y}_j|}\}$ be a $\mathcal{Y}_j$-slicing of $\{A^{(\mathcal{X})}_x \colon x \in \mathcal{X}\}$ and let $A_{C_{j,k}}$ be the $|\mathcal{X}| \times d_{\mathcal{X}}$-matrix with x-th row equal to $(A^{(\mathcal{X})}_x)^\top$ when $x \in C_{j,k}$ and equal to a row of zeros otherwise. Let $A_{C_j} = (A_{C_{j,1}} | \cdots | A_{C_{j,|\mathcal{Y}_j|}}) \in \mathbb{R}^{|\mathcal{X}| \times |\mathcal{Y}_j| d_{\mathcal{X}}}$ and $d = \sum_{j\in[m]} |\mathcal{Y}_j| d_{\mathcal{X}}$.

Theorem 17 *On each region of linearity, the tropical morphism Φ is the linear map $\mathbb{R}^d \to \mathrm{RBM}^{\mathrm{tropical}}_{\mathcal{X},\mathcal{Y}}$ represented by the $|\mathcal{X}| \times d$-matrix*

$$\mathcal{A} = (A_{C_1} | \cdots | A_{C_m}),$$

modulo constant functions. In particular, $\dim(\mathrm{RBM}^{\mathrm{tropical}}_{\mathcal{X},\mathcal{Y}}) + 1$ *is the maximum rank of $\mathcal{A}$ over all possible collections of slicings $C_1, \ldots, C_m$.*

Proof Again use the homogeneous version of the matrix $A^{(\mathcal{X},\mathcal{Y})}$ as in the proof of Proposition 6; this will not affect the rank of $\mathcal{A}$. Let $\theta_{h_j} = (\theta_{\{j,i\},(h_j,x_i)})_{i\in[n],x_i\in\mathcal{X}_i}$ and let A_{h_j} denote the submatrix of $A^{(\mathcal{X},\mathcal{Y})}$ containing the rows with indices $\{\{j, i\}, (h_j, x_i) \colon i \in [n], x_i \in \mathcal{X}_i\}$. For any given $v \in \mathcal{X}$ we have

$$\max\Big\{\big\langle \theta, A^{(\mathcal{X},\mathcal{Y})}_{(v,h)} \big\rangle \colon h \in \mathcal{Y}\Big\} = \sum_{j\in[m]} \max\Big\{\big\langle \theta_{h_j}, A_{h_j}(v, h_j) \big\rangle \colon h_j \in \mathcal{Y}_j\Big\},$$

from which the claim follows. ∎

In the following we evaluate the maximum rank of the matrix $\mathcal{A}$ for various choices of $\mathcal{X}$ and $\mathcal{Y}$ by examining good slicings. We focus on slicings by parallel hyperplanes.

Lemma 18 *For any $x^* \in \mathcal{X}$ and $0 < k < n$ the affine hull of the set $\{A_x^{(\mathcal{X})} : d_H(x, x^*) = k\}$ has dimension $\sum_{i\in[n]}(|\mathcal{X}_i| - 1) - 1$.*

Proof Without loss of generality let $x^* = (0, \ldots, 0)$. The set $\mathcal{Z}^k := \{A_x^{(\mathcal{X})} : d_H(x, x^*) = k\}$ is the intersection of $\{A_x^{(\mathcal{X})} : x \in \mathcal{X}\}$ with the hyperplane $H^k := \{z : \langle \mathbb{1}, z\rangle = k + 1\}$. Now note that the two vertices of an edge of $Q_{\mathcal{X}}$ either lie in the same hyperplane H^l, or in two adjacent parallel hyperplanes H^l and H^{l+1}, with $l \in \mathbb{N}$. Hence the hyperplane H^k does not slice any edges of $Q_{\mathcal{X}}$ and $\operatorname{conv}(\mathcal{Z}^k) = Q_{\mathcal{X}} \cap H^k$. The set $\mathcal{Z}^k$ is not contained in any proper face of $Q_{\mathcal{X}}$ and hence $\operatorname{conv}(\mathcal{Z}^k)$ intersects the interior of $Q_{\mathcal{X}}$. Thus $\dim(\operatorname{conv}(\mathcal{Z}^k)) = \dim(Q_{\mathcal{X}}) - 1$, as was claimed. ■

Lemma 18 implies the following.

Corollary 19 *Let $x \in \mathcal{X}$, and $2k-3 \leq n$. There is a slicing $C_1 = \{C_{1,1}, \ldots, C_{1,k}\}$ of $\mathcal{X}$ by $k-1$ parallel hyperplanes such that $\cup_{l=1}^{k-1} C_{1,l} = B_x(2k-3)$ is the Hamming ball of radius $2k-3$ centered at x and the matrix $A_{C_1} = (A_{C_{1,1}}|\cdots|A_{C_{1,k-1}})$ has full rank.*

Recall that $\mathfrak{A}_{\mathcal{X}}(d)$ denotes the maximal cardinality of a subset of $\mathcal{X}$ of minimum Hamming distance at least d. When $\mathcal{X} = \{0, 1, \ldots, q-1\}^n$ we write $\mathfrak{A}_q(n, d)$. Let $\mathfrak{K}_{\mathcal{X}}(d)$ denote the minimal cardinality of a subset of $\mathcal{X}$ with covering radius d.

Proposition 20 (Binary visible units) *Let $\mathcal{X} = \{0,1\}^n$ and $|\mathcal{Y}_j| = s_j$ for all $j \in [m]$. If $\mathcal{X}$ contains m disjoint Hamming balls of radii $2s_j - 3$, $j \in [m]$ whose complement has full rank, then $\mathrm{RBM}_{\mathcal{X},\mathcal{Y}}^{\mathrm{tropical}}$ has the expected dimension, $\min\{\sum_{j\in[m]}(s_j - 1)(n+1) + n, 2^n - 1\}$.*

In particular, if $\mathcal{X} = \{0,1\}^n$ and $\mathcal{Y} = \{0, 1, \ldots, s-1\}^m$ with $m < \mathfrak{A}_2(n, d)$ and $d = 4(s-1) - 1$, then $\mathrm{RBM}_{\mathcal{X},\mathcal{Y}}$ has the expected dimension. It is known that $\mathfrak{A}_2(n, d) \geq 2^{n - \lceil \log_2(\sum_{j=0}^{d-2} \binom{n-1}{j})\rceil}$.

Proposition 21 (Binary hidden units) *Let $\mathcal{Y} = \{0,1\}^m$ and $\mathcal{X}$ be arbitrary.*

- *If $m+1 \leq \mathfrak{A}_{\mathcal{X}}(3)$, then $\mathrm{RBM}_{\mathcal{X},\{0,1\}^m}^{\mathrm{tropical}}$ has dimension $(1+m)(1+\sum_{i\in[n]}(|\mathcal{X}_i|-1)) - 1$.*
- *If $m+1 \geq \mathfrak{K}_{\mathcal{X}}(1)$, then $\mathrm{RBM}_{\mathcal{X},\{0,1\}^m}^{\mathrm{tropical}}$ has dimension $|\mathcal{X}| - 1$.*

Let $\mathcal{Y} = \{0,1\}^m$ and $\mathcal{X} = \{0, 1, \ldots, q-1\}^n$, where q is a prime power.

- *If $m+1 \leq q^{n - \lceil \log_q(1+(n-1)(q-1)+1)\rceil}$, then $\mathrm{RBM}_{\mathcal{X},\mathcal{Y}}^{\mathrm{tropical}}$ has dimension $(1+m)(1+\sum_{i\in[n]}(|\mathcal{X}_i|-1)) - 1$.*
- *If $n = (q^r - 1)/(q-1)$ for some $r \geq 2$, then $\mathcal{A}_{\mathcal{X}}(3) = \mathfrak{K}_{\mathcal{X}}(1)$, and $\mathrm{RBM}_{\mathcal{X},\mathcal{Y}}^{\mathrm{tropical}}$ has the expected dimension for any m.*

In particular, when all units are binary and $m < 2^{n - \lceil \log_2(n+1)\rceil}$, then $\mathrm{RBM}_{\mathcal{X},\mathcal{Y}}$ has the expected dimension; this was shown in (Cueto et al., 2010).

Proposition 22 (Arbitrary sized units) *If $\mathcal{X}$ contains m disjoint Hamming balls of radii $2|\mathcal{Y}_1|-3, \dots, 2|\mathcal{Y}_m|-3$, and the complement of their union has full rank, then $\mathrm{RBM}^{\mathrm{tropical}}_{\mathcal{X},\mathcal{Y}}$ has the expected dimension.*

Proof Propositions 20, 21, and 22 follow from Theorem 17 and Corollary 19 together with the following explicit bounds on $\mathfrak{A}$ by Gilbert (1952); Varshamov (1957):

$$\mathfrak{A}_q(n,d) \geq \frac{q^n}{\sum_{j=0}^{d-1} \binom{n}{j}(q-1)^j}.$$

If q is a prime power, then $\mathfrak{A}_q(n,d) \geq q^k$, where k is the largest integer with $q^k < \frac{q^n}{\sum_{j=0}^{d-2}\binom{n-1}{j}(q-1)^j}$. In particular, $\mathfrak{A}_2(n,3) \geq 2^k$, where k is the largest integer with $2^k < \frac{2^n}{(n-1)+1} = 2^{n-\log_2(n)}$, i.e., $k = n - \lceil \log_2(n+1) \rceil$. ■

Example 5 Many results in coding theory can now be translated directly to statements about the dimension of discrete RBMs. Here is an example. Let $\mathcal{X} = \{1,2,\dots,s\} \times \{1,2,\dots,s\} \times \{1,2,\dots,t\}$, $s \leq t$. The minimum cardinality of a code $C \subseteq \mathcal{X}$ with covering-radius one equals $\mathfrak{K}_{\mathcal{X}}(1) = s^2 - \left\lfloor \frac{(3s-t)^2}{8} \right\rfloor$ if $t \leq 3s$, and $\mathfrak{K}_{\mathcal{X}}(1) = s^2$ otherwise (Cohen et al., 2005, Theorem 3.7.4). Hence $\mathrm{RBM}^{\mathrm{tropical}}_{\mathcal{X},\{0,1\}^m}$ has dimension $|\mathcal{X}|-1$ when $m+1 \geq s^2 - \left\lfloor \frac{(3s-t)^2}{8} \right\rfloor$ and $t \leq 3s$, and when $m+1 \geq s^2$ and $t > 3s$.

8. Discussion

In this note we study the representational power of RBMs with discrete units. Our results generalize a diversity of previously known results for standard binary RBMs and naïve Bayes models. They help contrasting the geometric-combinatorial properties of distributed products of experts versus non-distributed mixtures of experts.

We estimate the number of hidden units for which discrete RBM models can approximate any distribution to any desired accuracy, depending on the cardinalities of their units' state spaces. This analysis shows that the maximal approximation error increases at most logarithmically with the total number of visible states and decreases at least logarithmically with the sum of the number of states of the hidden units. This observation could be helpful, for example, in designing a penalty term to allow comparison of models with differing numbers of units. It is worth mentioning that the submodels of discrete RBMs described in Theorem 12 can be used not only to estimate the maximal model approximation errors, but also the expected model approximation errors given a prior of target distributions on the probability simplex (Montúfar and Rauh, 2014). In future work it would be interesting to study the statistical approximation errors of discrete RBMs and to complement the theory by an empirical evaluation.

The combinatorics of tropical discrete RBMs allows us to relate the dimension of discrete RBM models to the solutions of linear optimization problems and slicings of convex support polytopes by normal fans of simplices. We use this to show that the model $\mathrm{RBM}_{\mathcal{X},\mathcal{Y}}$ has the expected dimension for many choices of $\mathcal{X}$ and $\mathcal{Y}$, but not for all choices. We

based our explicit computations of the dimension of RBMs on slicings by collections of parallel hyperplanes, but more general classes of slicings may be considered. The same tools presented in this paper can be used to estimate the dimension of a general class of models involving interactions within layers, defined as Kronecker products of hierarchical models (Montúfar and Morton, 2013). We think that the geometric-combinatorial picture of discrete RBMs developed in this paper may be helpful in solving various long standing theoretical problems in the future, for example: What is the exact dimension of naïve Bayes models with general discrete variables? What is the smallest number of hidden variables that make an RBM a universal approximator? Do binary RBMs always have the expected dimension?

Acknowledgments

We are grateful to the ICLR 2013 community for very valuable comments. This work was accomplished while G.M. was with the Department of Mathematics at the Pennsylvania State University in 2012 and 2013 and in part during his stays at the Max Planck Institute for Mathematics in the Sciences in September and October 2012. This work was supported in part by DARPA grant FA8650-11-1-7145.

References

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

N. Ay and A. Knauf. Maximizing multi-information. *Kybernetika*, 42(5):517–538, 2006.

Y. Bengio. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2(1):1–127, 2009.

Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. Greedy layer-wise training of deep networks. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 153–160. MIT Press, Cambridge, MA, 2007.

M. Á. Carreira-Perpiñán and G. E. Hinton. On contrastive divergence learning. In *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics*, AISTATS '05, 2005.

M. V. Catalisano, A. V. Geramita, and A. Gimigliano. Secant varieties of $\mathbb{P}^1 \times \cdots \times \mathbb{P}^1$ (n-times) are not defective for $n \geq 5$. *Journal of Algebraic Geometry*, 20:295–327, 2011.

G. Cohen, I. Honkala, S. Litsyn, and A. Lobstein. *Covering Codes*. North-Holland Mathematical Library. Elsevier Science, 2005.

M. A. Cueto, J. Morton, and B. Sturmfels. Geometry of the restricted Boltzmann machine. In M. Viana and H. Wynn, editors, *Algebraic methods in statistics and probability II, AMS Special Session*, volume 2. American Mathematical Society, 2010.

G. E. Dahl, R. P. Adams, and H. Larochelle. Training restricted Boltzmann machines on word observations. In John Langford and Joelle Pineau, editors, *Proceedings of the 29th International Conference on Machine Learning*, ICML '12, pages 679–686, New York, NY, USA, July 2012. Omnipress.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

J. Draisma. A tropical approach to secant dimensions. *Journal of Pure and Applied Algebra*, 212(2):349–363, 2008.

Y. Freund and D. Haussler. Unsupervised learning of distributions of binary vectors using 2-layer networks. In J. E. Moody, S. J. Hanson, and R. Lippmann, editors, *Advances in Neural Information Processing Systems 4*, NIPS '91, pages 912–919. Morgan Kaufmann, 1991.

E. N. Gilbert. A comparison of signalling alphabets. *Bell System Technical Journal*, 31: 504–522, 1952.

G. E. Hinton. Products of experts. In *Proceedings of the 9th International Conference on Artificial Neural Networks*, volume 1 of *ICANN '99*, pages 1–6, 1999.

G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002.

G. E. Hinton. A practical guide to training restricted Boltzmann machines, version 1. Technical report, UTML2010-003, University of Toronto, 2010.

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

N. Le Roux and Y. Bengio. Representational power of restricted Boltzmann machines and deep belief networks. *Neural Computation*, 20(6):1631–1649, 2008.

P. M. Long and R. A. Servedio. Restricted Boltzmann machines are hard to approximately evaluate or simulate. In J. Fürnkranz and T. Joachims, editors, *Proceedings of the 27th International Conference on Machine Learning*, ICML '10, pages 703–710. Omnipress, 2010.

D. Lowd and P. Domingos. Naive Bayes models for probability estimation. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML, pages 529–536. ACM Press, 2005.

T. K. Marks and J. R. Movellan. Diffusion networks, products of experts, and factor analysis. In *Proceedings of the 3rd International Conference Independent Component Analysis*, pages 481–485, 2001.

F. Matúš and N. Ay. On maximization of the information divergence from an exponential family. In *Proceedings of the 6th Workshop on Uncertainty Processing*, WUPES '03, pages 199–204. University of Economics, Prague, 2003.

R. Memisevic and G. E. Hinton. Learning to represent spatial transformations with factored higher-order Boltzmann machines. *Neural Computation*, 22(6):1473–1492, June 2010.

G. Montúfar. Mixture decompositions of exponential families using a decomposition of their sample spaces. *Kybernetika*, 49(1):23–39, 2013.

G. Montúfar and N. Ay. Refinements of universal approximation results for deep belief networks and restricted Boltzmann machines. *Neural Computation*, 23(5):1306–1319, 2011.

G. Montúfar and J. Morton. Geometry of hidden-visible products of statistical models. 2013. In preparation.

G. Montúfar and J. Morton. When does a mixture of products contain a product of mixtures? *SIAM Journal on Discrete Mathematics*, 29:321–347, 2015.

G. Montúfar and J. Rauh. Scaling of model approximation errors and expected entropy distances. *Kybernetika*, 50(2):234–245, 2014. Special issue of the 9th Workshop on Uncertainty Processing (WUPES '12).

G. Montúfar, J. Rauh, and N. Ay. Expressive power and approximation errors of restricted Boltzmann machines. In J. Shawe-Taylor, R.S. Zemel, P. Bartlett, F. C. N. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, NIPS '11, pages 415–423, 2011.

S. Osindero and G. E. Hinton. Modeling image patches with a directed hierarchy of Markov random fields. In J. C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, NIPS '07, pages 1121–1128. MIT Press, Cambridge, MA, 2008.

L. Pachter and B. Sturmfels. Tropical geometry of statistical models. *Proceedings of the National Academy of Sciences of the United States of America*, 101(46):16132–16137, November 2004.

M. Ranzato, A. Krizhevsky, and G. E. Hinton. Factored 3-Way Restricted Boltzmann Machines For Modeling Natural Images. In *Proceedings 13th International Conference on Artificial Intelligence and Statistics*, AISTATS '10, pages 621–628, 2010.

W. E. Roth. On direct product matrices. *Bulletin of the American Mathematical Society*, 40:461–468, 1934.

R. Salakhutdinov, A. Mnih, and G. E. Hinton. Restricted Boltzmann machines for collaborative filtering. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 791–798, New York, NY, USA, 2007. ACM.

T. J. Sejnowski. Higher-order Boltzmann machines. In *Neural Networks for Computing*, pages 398–403. American Institute of Physics, 1986.

P. Smolensky. Information processing in dynamical systems: foundations of harmony theory. In *Symposium on Parallel and Distributed Processing*, 1986.

T. Tran, D. Phung, and S. Venkatesh. Mixed-variate restricted Boltzmann machines. In *Proceedings of the 3rd Asian Conference on Machine Learning*, ACML '11, pages 213–229, 2011.

R. R. Varshamov. Estimate of the number of signals in error correcting codes. *Doklady Akademii Nauk SSSR*, 117:739–741, 1957.

M. Welling, M. Rosen-Zvi, and G. E. Hinton. Exponential family harmoniums with an application to information retrieval. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, NIPS '04, pages 1481–1488. MIT Press, Cambridge, MA, 2005.

Journal of Machine Learning Research 16 (2015) 673-712 Submitted 11/12; Revised 7/14; Published 4/15

Evolving GPU Machine Code

Cleomar Pereira da Silva CLEOMAR.SILVA@IFC-VIDEIRA.EDU.BR
Department of Electrical Engineering
Pontifical Catholic University of Rio de Janeiro (PUC-Rio)
Rio de Janeiro, RJ 22451-900, Brazil
Department of Education Development
Federal Institute of Education, Science and Technology - Catarinense (IFC)
Videira, SC 89560-000, Brazil

Douglas Mota Dias DOUGLASM@ELE.PUC-RIO.BR
Department of Electrical Engineering
Pontifical Catholic University of Rio de Janeiro (PUC-Rio)
Rio de Janeiro, RJ 22451-900, Brazil

Cristiana Bentes CRIS@ENG.UERJ.BR
Department of Systems Engineering
State University of Rio de Janeiro (UERJ)
Rio de Janeiro, RJ 20550-013, Brazil

Marco Aurélio Cavalcanti Pacheco MARCO@ELE.PUC-RIO.BR
Department of Electrical Engineering
Pontifical Catholic University of Rio de Janeiro (PUC-Rio)
Rio de Janeiro, RJ 22451-900, Brazil

Leandro Fontoura Cupertino FONTOURA@IRIT.FR
Toulouse Institute of Computer Science Research (IRIT)
University of Toulouse
118 Route de Narbonne
F-31062 Toulouse Cedex 9, France

Editor: Una-May O'Reilly

Abstract

Parallel Graphics Processing Unit (GPU) implementations of GP have appeared in the literature using three main methodologies: (i) *compilation*, which generates the individuals in GPU code and requires compilation; (ii) *pseudo-assembly*, which generates the individuals in an intermediary assembly code and also requires compilation; and (iii) *interpretation*, which interprets the codes. This paper proposes a new methodology that uses the concepts of quantum computing and directly handles the GPU machine code instructions. Our methodology utilizes a probabilistic representation of an individual to improve the global search capability. In addition, the evolution in machine code eliminates both the overhead of compiling the code and the cost of parsing the program during evaluation. We obtained up to 2.74 trillion GP operations per second for the 20-bit Boolean Multiplexer benchmark. We also compared our approach with the other three GPU-based acceleration methodologies implemented for quantum-inspired linear GP. Significant gains in performance were obtained.

Keywords: genetic programming, graphics processing units, machine code.

1. Introduction

Genetic programming (GP) is a metaheuristic method to automatically generate computer programs or key subcomponents (Banzhaf et al., 1997; Koza, 1992; Poli et al., 2008). Its functionality is based on the Darwinian principle of natural selection, in which a population of computer programs, or individuals, is maintained and modified based on genetic variation. The individuals are then evaluated according to a fitness function to reach a better solution. GP has been successfully applied to a variety of problems, such as automatic design, pattern recognition, robotic control, data mining, and image analysis (Koza, 1992, 1994; Tackett, 1993; Busch et al., 2002; Harding and Banzhaf, 2008; Langdon, 2010a). However, the evaluation process is time consuming. The computational power required by GP is enormous, and high-performance techniques have been used to reduce the computation time (Andre and Koza, 1996; Salhi et al., 1998). GP parallelism can be exploited on two levels: multiple individuals can be evaluated simultaneously, or multiple fitness cases for one individual can be evaluated in parallel. These approaches have been implemented in multiprocessor machines and computer clusters (Page et al., 1999; Turton et al., 1996; Bennett III et al., 1999).

The recent emergence of general-purpose computing on Graphics Processing Units (GPUs) has provided the opportunity to significantly accelerate the execution of many costly algorithms, such as GP algorithms. GPUs have become popular as accelerators due to their high computational power, low cost, impressive floating-point capabilities, and high memory bandwidth. These characteristics make them attractive platforms to accelerate GP computations, as GP has a fine-grained parallelism that is suitable for GPU computation.

The power of the GPU to accelerate GP has been exploited in previous studies. We divide these efforts into three main methodologies: (i) *compilation* (Chitty, 2007; Harding and Banzhaf, 2007, 2009; Langdon and Harman, 2010); (ii) *pseudo-assembly* (Cupertino et al., 2011; Pospichal et al., 2011; Lewis and Magoulas, 2011); and (iii) *interpretation* (Langdon and Banzhaf, 2008a; Langdon and Harrison, 2008; Robilliard et al., 2009; Wilson and Banzhaf, 2008). In the *compilation* methodology, each evolved program, or GP individual, is compiled for the GPU machine code and then evaluated in parallel on the GPU. In the *pseudo-assembly* methodology, the individuals are generated in the pseudo-assembly code of the GPU, and a just-in-time (JIT) compilation is performed for each individual to generate the GPU machine code, which is evaluated in parallel on the GPU. In the *interpreter* methodology, an interpreter that can run programs immediately is used. The individuals are evaluated in parallel on the GPU.

These methodologies have been used with varying levels of success, and they have different advantages and disadvantages. In the *compilation* methodology, the GPU's fine-grain parallelism can be exploited by evaluating multiple individuals and multiple fitness cases simultaneously. However, the time spent compiling each GP individual influences the performance results considerably, making the GPU compiler decidedly slow. The compilation process in a GPU involves a series of steps. When GP needs to evaluate millions of programs, spending a few seconds to compile a single CUDA program becomes a large obstacle to producing a solution within a reasonable period of time. The *pseudo-assembly* methodology can also exploit multiple individuals and multiple fitness case evaluations in parallel. A pseudo-assembly code can be compiled several hundred times faster than an original GPU

code, allowing large data sets to be considered. Nevertheless, the programs still need to be compiled, and the compilation time must be considered as part of the overall GP process. The *interpreter* methodology differs from the *compilation* methodology in that the interpreter is compiled once and reused millions of times. This approach eliminates the compilation overhead but includes the cost of parsing the evolved program. The *interpreter* methodology typically works well for shorter programs and smaller training cases.

In this work, we propose a new methodology for using GPUs in the GP evolution process. We used a quantum-inspired evolutionary algorithm (QEA) that handles the instructions of the GPU machine code directly. QEAs represent one of the most recent advances in evolutionary computation (Zhang, 2011). QEAs are based on quantum mechanics, particularly the concepts of the quantum bit and the superposition of states. QEAs can represent diverse individuals in a probabilistic manner. By this mechanism, QEAs offer an evolutionary mechanism that is different and, in some situations, more effective than traditional evolutionary algorithms. The quantum probabilistic representation reduces the number of chromosomes required to guarantee adequate search diversity. In addition, the use of quantum interference provides an effective approach to achieve fast convergence to the best solution due to the inclusion of an individual's past history. It offers a guide for the population of individuals that helps to exploit the current solution's neighborhood.

Our methodology is called GPU machine code genetic programming, **GMGP**, and is based on linear genetic programming (LGP) (Nordin, 1998; Brameier and Banzhaf, 2007; Oltean et al., 2009). In LGP, each program is a linear sequence of instructions. LGP is the most appropriate for machine code programs, as computer architectures require programs to be provided as linear sequences. Computers do not naturally run tree-shaped programs. Tree-based GP must employ compilers or interpreters (Poli et al., 2008).

GMGP performs the evolution by modifying the GPU machine code, thus eliminating the time spent compiling the individuals while also avoiding the interpretation overhead. The individuals are generated on the CPU, and the individuals are evaluated in parallel on the GPU. The evaluation process is performed with a high level of parallelism: individuals are processed in parallel, and the fitness cases are simultaneously evaluated in parallel. Figure 1 illustrates the GPU-accelerated GP methodologies.

We compared our quantum-inspired methodology with the previous attempts to accelerate GP using GPUs. Our comparison considered the *compilation*, *pseudo-assembly*, and *interpretation* methodologies. We implemented these three methodologies to conform with linear GP and quantum-inspired algorithms, and to provide fair comparisons. GMGP outperformed all of these methodologies. The gains over *compilation* and *pseudo-assembly* originated from the elimination of the compilation time. The gains over *interpretation* originated from two sources. The first was the lack of the on-the-fly interpretation overhead. The second was the high number of comparison and jump instructions required by the interpreter, which produces serialization in the GPU execution. The main obstacle faced by GMGP was that the GPU machine code is proprietary, and the GPU's manufacturers do not provide any documentation for it. To solve this problem, we had to use reverse engineering to disassemble a series of GPU binary codes and determine the opcodes of the relevant instructions.

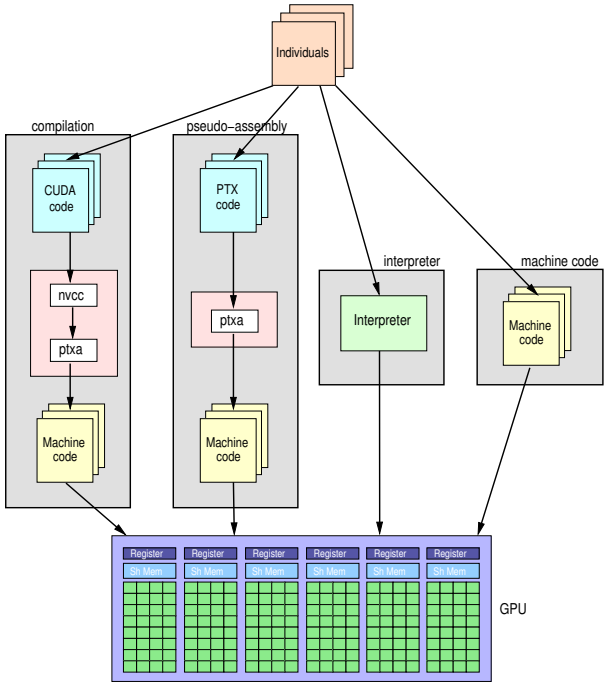


Figure 1: The different GP methodologies for GPU, considering the Nvidia technology. In the compilation methodology, a CUDA kernel is generated from each individual. The kernels are compiled in two main steps using the *nvcc* and *ptxas* compilers. In the pseudo-assembly methodology, pseudo-assembly codes (PTX) are generated from each individual and compiled using the *ptxas* compiler. In the interpreter methodology, each individual's information is used by the interpreter to execute the program. The proposed machine code methodology generates a machine code program directly from each individual.

2. Related Work

Several approaches to accelerate GP on GPUs have been proposed in the literature. Harding and Banzhaf (2007) and Chitty (2007) were the first to present GP implementations on a GPU. Both works proposed compiler methodologies using tree-based GP. They obtained modest performance gains when small fitness cases were tested due to the overhead of transferring data to the GPU. Considerable performance gains were obtained for larger problems and when the compiled GP program was run many times.

Langdon and Banzhaf (2008a) were the first to propose an interpreter methodology. Their methodology used a tree-based GP and evaluated the entire population at once. Parallelism was exploited at the individual level, whereas the fitness cases were processed sequentially. Their technique was called the SIMD interpreter for GP, and they used conditional instructions to select opcodes, which can increase the overhead with the size of

the function set. The experimental results indicated moderate speedups but demonstrated performance gains even for very small programs. The same GPU SIMD interpreter was used by Langdon and Harrison (2008), who successfully applied GP to predict the breast cancer survival rate beyond ten years.

Robilliard et al. (2009) also studied the interpreter methodology, with a focus on avoiding the overhead of conditional instructions when interpreting the entire population at once. They proposed an interpreter that evaluates each GP individual on a different thread block. Each thread block was mapped to a different GPU multiprocessor during execution, avoiding branches. Inside the thread block, all threads executed the same instruction over different data subsets. Their results indicated performance gains compared to the methodology proposed by Langdon and Banzhaf (2008a).

Harding and Banzhaf (2009) studied the compilation methodology. A cluster of GPUs was used to alleviate the program compilation overhead. The focus was on processing very large data sets by using the cluster nodes to compile the GPU code and execute the programs. Different combinations of compilation and execution nodes could be used. The project was developed to run on a multi-platform Windows/Linux cluster and used low-end GPUs. Speedups were obtained for very large data sets. However, the use of high-end GPUs did not necessarily lead to better results, as the primary bottleneck remained in the compilation phase.

Langdon and Harman (2010) used the compilation methodology to automatically create an Nvidia CUDA kernel. Numerous simplifications were employed, such as not evolving the shared memory and threading information. The best evolved parallel individual was capable of correct calculations, proving that it was possible to elaborate a methodology to evolve parallel code. However, it was not possible to automatically verify the speedup obtained compared to the sequential CPU version, and the compilation still remained the bottleneck.

Wilson and Banzhaf (2008) implemented an LGP for GPU using the interpreter methodology on a video game console. In a previous work (Cupertino et al., 2011), we proposed a pseudo-assembly methodology, a modified LGP for GPU, called quantum-inspired linear genetic programming on a general-purpose graphics processing unit (QILGP3U). The individual was created in the Nvidia pseudo-assembly code, PTX, and compiled for evaluation through JIT. Dynamic or JIT compilation is performed in runtime and transformed the assembly code to machine code during the execution of the program. Several compilation phases were eliminated, and significant speedups were achieved for large data sets. Pospichal et al. (2011) also proposed a pseudo-assembly methodology with the evolution of PTX code using a grammar-based GP that ran entirely on the GPU.

The compilation time issue was addressed in a different manner by Lewis and Magoulas (2011). All population individuals were pre-processed to identify their similarities, and all of these similarities were grouped together. In this manner, repetitive compilation was eliminated, thus reducing the compilation time by a factor of up to 4.8.

To our knowledge, no prior work has evolved GPU programs by directly handling the GPU machine code itself.

3. Quantum Computing and Quantum-Inspired Algorithms

In a classical computer, a *bit* is the smallest information unit and can take a value of 0 or 1. In a quantum computer, the basic information unit is the *quantum bit*, called the *qubit*. A qubit can take the states $|0\rangle$ or $|1\rangle$ or a superposition of the two. This superposition of the two states is a linear combination of the states $|0\rangle$ and $|1\rangle$ and can be represented as follows:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \tag{1}$$

where $|\psi\rangle$ is the qubit state , α and β are complex numbers, and $|\alpha|^2$ and $|\beta|^2$ are the probabilities that the qubit collapses to state 0 or 1, respectively, based on its observation (i.e., measurement). The unitary normalization guarantees the following:

$$|\alpha|^2 + |\beta|^2 = 1 \mid \{\alpha,\beta\} \in \mathbb{C}. \tag{2}$$

The superposition of states provides quantum computers with an incomparable degree of parallelism. This parallelism, when properly exploited, allows computers to perform tasks that are unfeasible in classical computers due to the prohibitive computational time.

Although quantum computing is promising in terms of processing capacity, there is still no technology for the actual implementation of a quantum computer, and there are only a few complex quantum algorithms.

Moore and Narayanan (1995) proposed a new approach to exploit the quantum computing concepts. Instead of developing new algorithms for quantum computers or attempting to make their use feasible, they proposed the idea of *quantum-inspired computing*. This new approach aims to create classical algorithms (i.e., running on classical computers) that utilize quantum mechanics paradigms to improve their problem-solving performance. In particular, quantum-inspired evolutionary algorithms (QEAs) have recently become a subject of special interest in evolutionary computation. The linear superposition of states represented in a qubit allows QEA to represent diverse individuals probabilistically. QEAs belong to the class of estimation of distribution algorithms (EDAs) (Platel et al., 2009). The probabilistic mechanism provides QEAs with an evolutionary mechanism that has several advantages, such as global search capability and faster convergence and smaller population size than those of traditional evolutionary algorithms. These algorithms have already been successfully used to solve various problems, such as the knapsack problem (Han and Kim, 2002), ordering combinatorial optimization problems (Silveira et al., 2012), engineering optimization problems (Alfares and Esat, 2006), image segmentation (Talbi et al., 2007), and image registration (Draa et al., 2004). See Zhang (2011) for more examples of QEAs and their applications.

3.1 Multilevel Quantum Systems

Most quantum computing approaches use qubits encoded in two-level quantum systems. However, the candidate systems for encoding quantum information often have a more complex physical structure, with several directly accessible degrees of freedom (e.g., atoms, ions, photons). Quantum systems of d levels were recently studied, where the *qudit* is the quantum information unit, which may take any of d values or a superposition of d states (Lanyon et al., 2008).

4. Quantum-Inspired Linear Genetic Programming

The proposed quantum-inspired GP methodology for GPUs is based on the quantum-inspired linear genetic programming (QILGP) algorithm proposed by Dias and Pacheco (2013). QILGP evolves machine code programs for the Intel x86 platform. It uses floating point instructions and works with data from the main memory (m) and/or eight FPU registers ($ST(i) \mid i \in [0\,..\,7]$). The function set consists of addition, subtraction, multiplication, division, data transfer, trigonometric, and other arithmetic instructions. QILGP generates variable-sized programs by adding the `NOP` instruction to the instruction set. The code generation ignores any gene in which a `NOP` is present. Table 1 provides an example of a function set.

Each individual is represented by a linear sequence of machine code instructions. Each instruction can use one or zero arguments. The evaluation of a program requires the input data to be read from the main memory, which consists of the input variables of the problem and some optional constants supplied by the user. The input data are represented by a vector, such as

$$I = (V[0], V[1], 1, 2, 3)\,, \tag{3}$$

where $V[0]$ and $V[1]$ have the two input values of the problem (i.e., a fitness case) and 1, 2, and 3 are the three constant values.

The instructions are represented in QILGP by two *tokens*: the *function token* (FT), which represents the function, and the *terminal token* (TT), which represents the argument of the function. Each function has a single terminal. When a function has no terminal, its corresponding token value is ignored. Each token is an integer value that represents an index to the function set or terminal set.

4.1 Representation

QILGP is based on the following entities: the *quantum individual*, which represents the superposition of all possible programs for the defined search space, and the *classical individual* (or *individual*), which represents the machine code program coded in the token values. A classical individual represents an individual of a traditional linear GP. In the observation phase of QILGP, each quantum individual is observed to generate one classical individual.

4.2 Observation

The chromosome of a quantum individual is represented by a list of structures called *quantum genes*. The observation of a quantum individual comprises the observations of all of its chromosome genes. The observation process consists of randomly generating a value r $\{r \in \mathbb{R} \mid 0 \leq r \leq 1\}$ and searching for the interval in which r belongs in all possible states that the individual can represent. For example, the process of observing a quantum gene

Instruction	Operation	Arg.
NOP	No operation	-
FADD m	$ST(0) \leftarrow ST(0) + m$	m
FADD ST(0), ST(i)	$ST(0) \leftarrow ST(0) + ST(i)$	i
FADD ST(i), ST(0)	$ST(i) \leftarrow ST(i) + ST(0)$	i
FSUB m	$ST(0) \leftarrow ST(0) - m$	m
FSUB ST(0), ST(i)	$ST(0) \leftarrow ST(0) - ST(i)$	i
FSUB ST(i), ST(0)	$ST(i) \leftarrow ST(i) - ST(0)$	i
FMUL m	$ST(0) \leftarrow ST(0) \times m$	m
FMUL ST(0), ST(i)	$ST(0) \leftarrow ST(0) \times ST(i)$	i
FMUL ST(i), ST(0)	$ST(i) \leftarrow ST(i) \times ST(0)$	i
FXCH ST(i)	$ST(0) \leftrightarrows ST(i)$ (swap)	i
FDIV m	$ST(0) \leftarrow ST(0) \div m$	m
FDIV ST(0), ST(i)	$ST(0) \leftarrow ST(0) \div ST(i)$	i
FDIV ST(i), ST(0)	$ST(i) \leftarrow ST(i) \div ST(0)$	i
FABS	$ST(0) \leftarrow \lvert ST(0) \rvert$	-
FSQRT	$ST(0) \leftarrow \sqrt{ST(0)}$	-
FSIN	$ST(0) \leftarrow \sin ST(0)$	-
FCOS	$ST(0) \leftarrow \cos ST(0)$	-

Table 1: Functional description of the instructions. The first column presents the Intel x86 instructions. The second column presents the operations performed. The third column presents the argument of the instructions (m indexes memory positions, and i selects a register).

represented by 10 different states follows the function

$$T(r) = \begin{cases} 0 & \text{if } 0 \leq r < p'_0 \\ 1 & \text{if } p'_0 \leq r < p'_1 \\ 2 & \text{if } p'_1 \leq r < p'_2 \\ \vdots & \vdots \\ 9 & \text{if } p'_8 \leq r \leq p'_9, \end{cases} \tag{4}$$

where $\{r \in \mathbb{R} \mid 0 \leq r \leq 1\}$ is the randomly generated value with a uniform distribution and $T(r)$ returns the observed value for the token.

The observation process plays an important role in the quantum-inspired evolutionary algorithm. The quantum-inspired representation of a gene implies that the creation of each instruction follows a probabilistic distribution, where it is possible to represent the instructions that are more likely to be observed. Furthermore, the evolutionary algorithm can be fed with the results of the individual evaluations, and the superposition of states allows the probability values to be improved iteratively. The best classical individuals contribute to improving the probability values of the quantum individuals. This mechanism enables the algorithm to achieve better solutions with fewer evaluations.

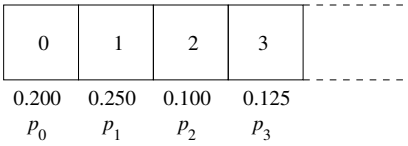


Figure 2: Illustration of a qudit implementation that represents Equation (6). Each state has an associated probability value and a token value. The observation process generates a random number r and selects one token based on the probability interval in which r fits.

QILGP is inspired by multilevel quantum systems (Lanyon et al., 2008), and uses the qudit as the basic information unit. This information can be described by a state vector of d levels, where d is the number of states in which the qudit can be measured. Accordingly, d represents the cardinality of the token. The state of a qudit is a linear superposition of d states and may be represented as follows:

$$|\psi\rangle = \sum_{i=0}^{d-1} \alpha_i |i\rangle , \tag{5}$$

where $|\alpha_i|^2$ is the probability that the qudit collapses to state i when observed.

For example, suppose that each instruction in Table 1 has a unique token value in $T = \{0,1,2,3,...\}$. Equation (6) provides the state of a function qudit (FQ) whose state is given as follows:

$$|\psi\rangle = \frac{1}{\sqrt{5}} |0\rangle + \frac{1}{\sqrt{4}} |1\rangle + \frac{1}{\sqrt{10}} |2\rangle + \frac{1}{\sqrt{8}} |3\rangle + \dots \tag{6}$$

The probability of measuring the `NOP` instruction (state $|0\rangle$) is $(1/\sqrt{5})^2 = 0.200$, for `FADD m` (state $|1\rangle$) is $(1/\sqrt{4})^2 = 0.250$, for `FADD ST(0),ST(i)` (state $|2\rangle$) is $(1/\sqrt{10})^2 = 0.100$, and so on. The qudit state of this example is implemented in a data structure as shown in Figure 2.

Figure 3 illustrates the creation of a classical gene by the observation of a quantum gene from an example based on Table 1 and the input vector $I = (V[0], V[1], 1, 2, 3)$ (Equation 3). This process can be explained by three basic steps, indicated by the numbered circles in Figure 3:

1. The FQ is observed, and the resulting value (e.g., 7) is assigned to the FT of this gene.

2. The FT value determines the terminal qudit (TQ) to be observed, as each instruction requires a different type of terminal: register or memory.

3. The TQ defined by the FT value is observed, and the resulting value (e.g., 1) is assigned to the TT of this gene.

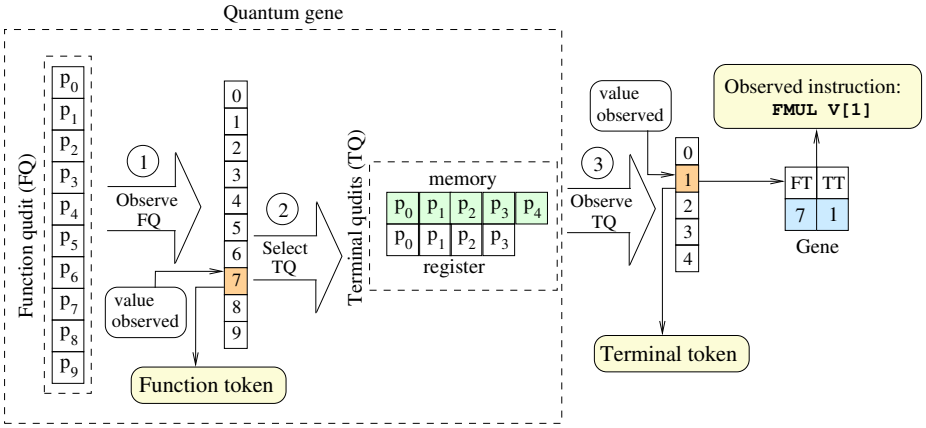


Figure 3: The creation of a classical gene from the observation of a quantum gene. The FQ is observed, and the token value selected is 7. The memory qudit is selected in the TQ. The TQ is observed, and the TT value selected is 1. The observed instruction in this example is `FMUL V[1]`, as '7' is the FT value for this instruction (Table 1), and '1' is the TT value that represents $V[1]$ in the input vector I defined by Equation (3).

4.3 Evaluation of a Classical Individual

This process begins with the generation of a machine code program from the classical individual under evaluation, where its chromosome is sequentially traversed, gene by gene and token by token (both FTs and TTs), to serially generate the program body machine code related to the classical individual. Then, the program is executed for all fitness cases of the problem (i.e., samples of the training data set).

For each fitness case, the value assigned as the result of the fitness case is zero ($V[0] \leftarrow 0$) when the instructions `FDIV` require division by zero or the instructions `FSQRT` require the calculation of the square root of a negative number.

4.4 Quantum Operator

The quantum operator of QILGP manipulates the probability p_i of a qudit, satisfying the normalization condition $\sum_{i=0}^{d-1} |\alpha_i|^2 = 1$, where d is the qudit cardinality and $|\alpha_i|^2 = p_i$. Operator P works in two main steps. First, it increases the given probability of a qudit as follows:

$$p_i \leftarrow p_i + s \times (1 - p_i), \tag{7}$$

where s is a parameter called *step size*, which can assume any real value between 0 and 1. The second step is to adjust the values of all of the probabilities of that qudit to satisfy the normalization condition. Thus, the operator modifies the state of a qudit by increasing p_i of a value that is directly proportional to s. The asymptotic behavior of p_i in Equation (7) indicates that the probability never reaches the unit value. This avoidance of unit probabilities is an important feature of this operator, as it avoids letting a probability

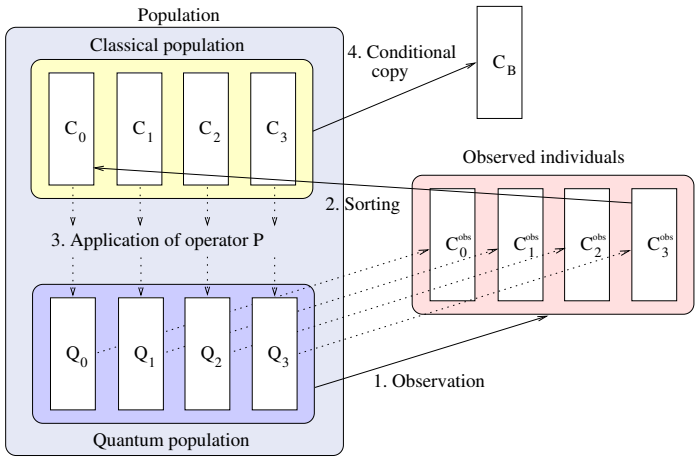


Figure 4: The four basic steps that characterize a generation of QILGP. With a population size of 4, the quantum individuals are observed and generate classical individuals. The classical individuals are sorted by their evaluations. The operator P is applied to each quantum individual, using the classical individual as the reference. The best classical individual evaluated thus far is kept in C_B.

cause the qudit to collapse, which could cause a premature convergence of the evolutionary search process.

QILGP has a hybrid population composed of a quantum population and classical population, both of which comprise M individuals. QILGP also has M auxiliary classical individuals C_i^{obs}, which result from observations of the quantum individuals Q_i, where $1 \leq i \leq M$.

4.5 Evolutionary Algorithm

Figure 4 illustrates the four basic steps that characterize a *generation* of QILGP, with a population size $M = 4$. The algorithm works as follows:

1. Each of M quantum individuals is observed once, resulting in M classical individuals C_i^{obs}.

2. The individuals of the classical population and the observed individuals (auxiliary) are jointly sorted by their evaluations, ordered from best to worst, from C_0 to C_{M-1}.

3. The operator P is applied to each quantum individual Q_i, taking their corresponding individual C_i in the classical population as a reference. Thus, at every new generation, the application of this operator increases the probability that the observations of the quantum individuals generate classical individuals more similar to the best individuals found thus far.

4. If any classical individual evaluated in the current generation is better than the best classical individual evaluated previously, a copy is stored in C_B, which keeps the best classical individual found by the algorithm thus far.

5. GPU Architecture

GPUs are highly parallel, many-core processors typically used as accelerators for a host system. They provide tremendous computational power and have proven to be successful for general-purpose parallel computing in a variety of application areas. Although different manufacturers have developed GPUs in recent years, we have opted for GPUs from Nvidia due to their flexibility and availability.

An Nvidia GPU consists of a set of streaming multiprocessors (SMs), each consisting of a set of GPU cores. The memory in the GPU is organized as follows: a large global memory with high latency; a very fast, low-latency on-chip shared memory for each SM; and a private local memory for each thread. Data communication between the GPU and CPU is conducted via the PCIe bus. The CPU and GPU have separate memory spaces, referred to as the host memory and device memory, and the GPU-CPU transfer time is limited by the speed of the PCIe bus.

5.1 Programming Model

The Nvidia programming model is CUDA (Computer Unified Device Architecture) (Nvidia, 2013). CUDA is a C-based development environment that allows the programmer to define special C functions, called *kernels*, which execute in parallel on the GPU by different threads. The GPU supports a large number of fine-grain threads. The threads are organized into a hierarchy of thread grouping. The threads are divided into a two- or three-dimensional *grid* of thread *blocks*. Each thread block is a two- or three-dimensional thread array. Thread blocks are executed on the GPU by assigning a number of blocks to be executed on a SM. Each thread in a thread block has a unique identifier, given by the built-in variables `threadIdx.x, threadIdx.y`, and `threadIdx.z`. Each thread block has an identifier that distinguishes its position in the grid, given by the built-in variables `blockIdx.x, blockIdx.y`, and `blockIdx.z`. The dimensions of the thread and thread block are specified at the time when the kernel is launched through the identifiers `blockDim` and `gridDim`, respectively.

All threads in a block are assigned to execute in the same SM. Hence, threads within one block can cooperate among themselves using synchronization primitives and shared memory. However, the number of threads within one block can exceed the number of cores in an SM, which requires a scheduling mechanism. The scheduling mechanism divides the block into *warps*. Each warp contains a fixed number of threads grouped by consecutive thread identifiers. The warp is executed on an SM in an implicit SIMD fashion, called SIMT (single instruction, multiple threads). Each core of an SM executes the same instruction simultaneously but on different data elements. However, the threads may logically follow a different control flow path and are free to branch. If some of the parallel threads choose a different execution path, called *code divergence*, their execution is serialized. In this case, the warp must be issued multiple times, one for each group of divergent threads. Thus, full efficiency is accomplished only when all of the threads in the warp follow the same execution path; otherwise, parallel efficiency can degrade significantly.

5.2 Compilation

The compilation of a CUDA program is performed through the following stages. First, the CUDA front end, *cudafe*, divides the program into the C/C++ host code and GPU device code. The host code is compiled with a regular C compiler, such as *gcc*. The device code is compiled using the CUDA compiler, *nvcc*, generating an intermediate code in an assembly language called PTX (Parallel Thread Execution). PTX is a human-readable, assembly-like low-level programming language for Nvidia GPUs that is compiled and hides many of the machine details. PTX has been fully documented by Nvidia. The PTX code is then translated to the GPU binary code, CUBIN, using the *ptxas* compiler.

Unlike the PTX language, whose documentation has been made public, the CUBIN format is proprietary, and no information has been made available by Nvidia. All of the work performed with CUBIN requires reverse engineering. In addition, the manufacturer provides only the most basic elements of the underlying hardware architecture, and there are apparently no plans to make more information public in the future.

6. GPU Machine Code Genetic Programming

Our GP methodology for GPUs is called GPU Machine Code Genetic Programming **GMGP**. It is a quantum-inspired LGP, based on QILGP, that evaluates the individuals on the GPU. The concept is to exploit the probabilistic representation of the individuals to achieve fast convergence and to parallelize the evaluation using the GPU machine code directly.

Before the evolution begins, the entire data set is transferred to the GPU global memory. In the first step, all of the classical individuals of one generation are created in the CPU in the same manner as in QILGP. Each classical individual is composed of tokens representing the instructions and arguments. For each individual, GMGP creates a GPU machine code kernel. These programs are then loaded to the GPU program memory and executed in parallel. The evaluation process in GMGP is performed with a high level of parallelism. We exploit the parallelism as follows: individuals are processed in parallel in different thread blocks, and data parallelism is exploited within each thread block, where each thread evaluates a different fitness case.

When the number of fitness cases is smaller than the number of threads in the block, we map one individual per block. For fitness cases greater than the number of threads per block, a two-dimensional grid is used, and each individual is mapped on multiple blocks. The individual is identified by the `blockIdx.y`, and the fitness case is identified by `(blockIdx.x * blockDim.x + threadIdx.x)`. To maintain all of the individual codes in a single GPU kernel, we use a set of IF statements to distinguish each individual. However, these IF statements do not introduce divergence in the kernel because all of the threads in each block follow the same execution path.

This methodology allows for the rapid evaluation of individuals. The GPU binary code is directly modified, thus avoiding the need to compile individuals. Regarding the machine code, our implementation is based on the Nvidia CUBIN code for the current Nvidia GPU architectures. Future Nvidia GPU machine code could be evolved using our methodology as long as the opcodes are known.

6.1 Function Set

GMGP is capable of evolving linear sequences of single precision floating point operations or linear sequences of Boolean operations. The function set of floating point operations is composed of addition, subtraction, multiplication, division, data transfer, trigonometric, and arithmetic instructions. The function set of Boolean operations is composed of AND, OR, NAND, NOR, and NOT. Table 2 provides the instruction set of the floating point operations, and Table 3 provides the instruction set of the Boolean operations. Each of these instructions has an opcode and one or two arguments. The argument can be a register or memory position. When it is a register, it varies from $R0$ to $R7$. When it is a memory position, it can be used to load input data or a constant value. The maximum number of inputs in GMGP is 256, and the maximum number of predefined constant values is 128. As an example, in Table 4, we present the CUBIN add instruction with all of the variations of its memory positions (X) and the eight auxiliary FPU registers ($Ri \mid i \in [0\,..\,7]$). Each CUBIN instruction variation with its arguments (constants or registers) has a different hexadecimal.

GMGP addresses only floating point and Boolean operations. Loops and jumps are not handled, as they are not common in the benchmark problems that we consider. However, GMGP could be extended to consider such problems, including mechanisms to restrict jumping to invalid positions and to avoid infinite loops.

Each evolved CUBIN program consists of three segments: *header*, *body*, and *footer*. The header and footer are the same for all individuals throughout the evolutionary process. They are optimized in the same manner as by the Nvidia compiler. These segments contain the following:

- *Header* – Loads the evaluation patterns from global memory to registers on the GPU and initializes eight registers with zero.
- *Body* – The evolved CUBIN code itself.
- *Footer* – Transfers $R0$ contents to the global memory, which is the default output of evolved programs, and then executes the exit instruction to terminate the program and return to the evolutionary algorithm main flow.

For each individual, the body of the program is assembled by stacking the hexadecimal code in the same order as the GP tokens have been read. There is no need for comparisons and branches within an individual code because the instructions are executed sequentially. Avoiding comparisons and branches is an important feature of GMGP. As explained before, GPUs are particularly sensitive to conditional branches.

We aggregate all program bodies of the same population into a single GPU kernel. The kernel has only one header and one footer, reducing the size of the population and thus decreasing the time to transfer the program to the GPU memory through the PCIe bus.

6.2 Machine Code Acquisition

We developed a semi-automatic procedure to acquire the GPU machine code instructions. Nvidia does not provide any documentation for its machine code.

CUDA	PTX	Description	A
		No operation	-
R0+=Xj ;	add.f32 R0, R0, Xj ;	$R(0) \leftarrow R(0) + X(j)$	j
R0+=Ri ;	add.f32 R0, R0, Ri ;	$R(0) \leftarrow R(0) + R(i)$	i
Ri+=R0 ;	add.f32 Ri, Ri, R0 ;	$R(i) \leftarrow R(i) + R(0)$	i
R0-=Xj ;	sub.f32 R0, R0, Xj ;	$R(0) \leftarrow R(0) - X(j)$	j
R0-=Ri ;	sub.f32 R0, R0, Ri ;	$R(0) \leftarrow R(0) - R(i)$	i
Ri-=R0 ;	sub.f32 Ri, Ri, R0 ;	$R(i) \leftarrow R(i) - R(0)$	i
R0*=Xj ;	mul.f32 R0, R0, Xj ;	$R(0) \leftarrow R(0) \times X(j)$	j
R0*=Ri ;	mul.f32 R0, R0, Ri ;	$R(0) \leftarrow R(0) \times R(i)$	i
Ri*=R0 ;	mul.f32 Ri, Ri, R0 ;	$R(i) \leftarrow R(i) \times R(0)$	i
R0/=Xj ;	div.full.f32 R0, R0, Xj ;	$R(0) \leftarrow R(0) \div X(j)$	j
R0/=Ri ;	div.full.f32 R0, R0, Ri ;	$R(0) \leftarrow R(0) \div R(i)$	i
Ri/=R0 ;	div.full.f32 Ri, Ri, R0 ;	$R(i) \leftarrow R(i) \div R(0)$	i
R8=R0;R0=Ri;Ri=R8;	mov.f32 R8, R0 ; mov.f32 R0, Ri ; mov.f32 Ri, R8 ;	$R(0) \overset{\leftarrow}{\rightarrow} R(i)$ (swap)	i
R0=abs(R0) ;	abs.f32 R0, R0 ;	$R(0) \leftarrow \lvert R(0) \rvert$	-
R0=sqrt(R0) ;	sqrt.approx.f32 R0, R0 ;	$R(0) \leftarrow \sqrt{R(0)}$	-
R0=__sinf(R0) ;	sin.approx.f32 R0, R0 ;	$R(0) \leftarrow \sin R(0)$	-
R0=__cosf(R0) ;	cos.approx.f32 R0, R0 ;	$R(0) \leftarrow \cos R(0)$	-

Table 2: Functional description of the single precision floating point instructions. The first column presents the CUDA command; the second presents the PTX instruction; the third describes the action performed; and the fourth column presents the argument for the instruction (j indexes memory positions, and i selects a register). The last two instructions, __sinf and __cosf, are fast_math instructions, which are less accurate but faster versions of sinf and cosf.

Our procedure creates a PTX program containing all of the PTX instructions listed in Tables 2 or 3. In this program, each instruction is embodied inside a loop, where the iteration count at the start of the loop is unknown, which prevents the *ptxas* compiler from removing instructions.

The PTX program is compiled, and the Nvidia *cuobjdump* tool is used to disassemble the binary code. The disassembled code contains the machine code of all instructions of the PTX program. The challenge is to remove the instructions that belong to each loop control, which is achieved by finding a pattern that repeats along the code. Once the loop controls are removed, each instruction of our instruction set is acquired.

The header and footer are obtained using the *xxd* tool from Linux, which converts binary programs into hex code and transforms the entire program into hexadecimal representation.

CUDA	PTX	Description	A
		No operation	-
R0=R0 & Xj ;	and.b32 R0, R0, Xj ;	$R(0) \leftarrow R(0) \wedge X(j)$	j
R0=R0 & Ri ;	and.b32 R0, R0, Ri ;	$R(0) \leftarrow R(0) \wedge R(i)$	i
Ri=Ri & R0 ;	and.b32 Ri, Ri, R0 ;	$R(i) \leftarrow R(i) \wedge R(0)$	i
R0=R0 — Xj ;	or.b32 R0, R0, Xj ;	$R(0) \leftarrow R(0) \vee X(j)$	j
R0=R0 — Ri ;	or.b32 R0, R0, Ri ;	$R(0) \leftarrow R(0) \vee R(i)$	i
Ri=Ri — R0 ;	or.b32 Ri, Ri, R0 ;	$R(i) \leftarrow R(i) \vee R(0)$	i
R0= ∼ (R0 & Xj) ;	and.b32 R0, R0, Xj ; not.b32 R0, R0 ;	$R(0) \leftarrow \overline{R(0) \wedge X(j)}$	j
R0= ∼ (R0 & Ri) ;	and.b32 R0, R0, Ri ; not.b32 R0, R0 ;	$R(0) \leftarrow \overline{R(0) \wedge R(i)}$	i
Ri= ∼ (Ri & R0) ;	and.b32 Ri, Ri, R0 ; not.b32 Ri, Ri ;	$R(i) \leftarrow \overline{R(i) \wedge R(0)}$	i
R0= ∼ (R0 — Xj) ;	or.b32 R0, R0, Xj ; not.b32 R0, R0 ;	$R(0) \leftarrow \overline{R(0) \vee X(j)}$	j
R0= ∼ (R0 — Ri) ;	or.b32 R0, R0, Ri ; not.b32 R0, R0 ;	$R(0) \leftarrow \overline{R(0) \vee R(i)}$	i
Ri= ∼ (Ri — R0) ;	or.b32 Ri, Ri, R0 ; not.b32 Ri, Ri ;	$R(i) \leftarrow \overline{R(i) \vee R(0)}$	i
R0= ∼ R0 ;	not.b32 R0, R0 ;	$R(0) \leftarrow \overline{R(0)}$	-

Table 3: Functional description of the Boolean instructions. The first column presents the CUDA command; the second presents the PTX instruction; the third describes the action performed; and the fourth column presents the argument for the instruction (j indexes memory positions, and i selects a register).

The header is the code that comes before the first instruction found, and the footer is the remaining code after the last instruction found.

With the header and footer, our procedure generates a different program to test each instruction acquired. This program contains a header, a footer, and one instruction. The program is executed, and the result is compared to an expected result that was previously computed on the CPU.

6.3 Evaluation Process

The GMGP methodology was explicitly designed to exploit the highly parallel capabilities of the GPU architecture. Because GMGP evaluates the entire population at once using two levels of parallelism, i.e., at the individual level and at the fitness case level, we expect our methodology to readily exploit future GPU architectures that are likely to have more processing cores than the recent releases. GMGP utilizes the independence of the fitness case execution and the ability to evaluate the individuals in parallel. In addition, this

CUBIN (hexadecimal representation)	Description	A
0x7e, 0x7c, 0x1c, **0x9**, 0x0, **0x80**, 0xc0, **0xe2**,		
0x7e, 0x7c, 0x1c, **0xa**, 0x0, **0x80**, 0xc0, **0xe2**,		
0x7d, 0x7c, 0x1c, 0x0, **0xfc, 0x81**, 0xc0, 0xc2,		
0x7d, 0x7c, 0x1c, 0x0, **0x0**, 0x82, 0xc0, 0xc2,		
0x7d, 0x7c, 0x1c, 0x0, **0x2**, 0x82, 0xc0, 0xc2,		
0x7d, 0x7c, 0x1c, 0x0, **0x4**, 0x82, 0xc0, 0xc2,	$R(0) \leftarrow R(0) + X(j)$	j
0x7d, 0x7c, 0x1c, 0x0, **0x5**, 0x82, 0xc0, 0xc2,		
0x7d, 0x7c, 0x1c, 0x0, **0x6**, 0x82, 0xc0, 0xc2,		
0x7d, 0x7c, 0x1c, 0x0, **0x7**, 0x82, 0xc0, 0xc2,		
0x7d, 0x7c, 0x1c, 0x0, **0x8**, 0x82, 0xc0, 0xc2,		
0x7d, 0x7c, 0x1c, **0x80, 0x8**, 0x82, 0xc0, 0xc2,		
0x7e, 0x7c, **0x9c, 0x0f**, 0x0, 0x80, 0xc0, 0xe2,		
0x7e, 0x7c, **0x1c, 0x0**, 0x0, 0x80, 0xc0, 0xe2,		
0x7e, 0x7c, **0x1c, 0x3**, 0x0, 0x80, 0xc0, 0xe2,		
0x7e, 0x7c, **0x9c, 0x3**, 0x0, 0x80, 0xc0, 0xe2,	$R(0) \leftarrow R(0) + R(i)$	i
0x7e, 0x7c, **0x1c, 0x4**, 0x0, 0x80, 0xc0, 0xe2,		
0x7e, 0x7c, **0x9c, 0x4**, 0x0, 0x80, 0xc0, 0xe2,		
0x7e, 0x7c, **0x1c, 0x5**, 0x0, 0x80, 0xc0, 0xe2,		
0x7e, 0x7c, **0x9c, 0x5**, 0x0, 0x80, 0xc0, 0xe2,		
0x7e, 0x7c, **0x9c, 0x0f**, 0x0, 0x80, 0xc0, 0xe2,		
0x2, 0x7c, **0x1c, 0x0**, 0x0, 0x80, 0xc0, 0xe2,		
0x1a, 0x7c, **0x1c, 0x3**, 0x0, 0x80, 0xc0, 0xe2,		
0x1e, 0x7c, **0x9c, 0x3**, 0x0, 0x80, 0xc0, 0xe2,	$R(i) \leftarrow R(i) + R(0)$	i
0x22, 0x7c, **0x1c, 0x4**, 0x0, 0x80, 0xc0, 0xe2,		
0x26, 0x7c, **0x9c, 0x4**, 0x0, 0x80, 0xc0, 0xe2,		
0x2a, 0x7c, **0x1c, 0x5**, 0x0, 0x80, 0xc0, 0xe2,		
0x2e, 0x7c, **0x9c, 0x5**, 0x0, 0x80, 0xc0, 0xe2,		

Table 4: Hexadecimal representation of the add GPU machine code instruction.

parallelization scheme avoids code divergence, as each thread in a block executes the same instruction over a different fitness case, and different individuals are executed by different thread blocks. Therefore, we are employing as much parallelism as possible for a population.

The evaluation process addresses the problems caused by execution errors, such as divisions by zero or square roots of negative numbers, which directly affect the fitness value of an evolved program. In both cases, the value assigned as the result is zero ($Ri \leftarrow 0$), which is the same approach adopted by the QILGP implementation (Dias and Pacheco, 2009).

7. Experiments and Results

In this section, we analyze the performance of GMGP compared with the other GP methodologies for GPUs. We describe the environment setup, the implementation of the other GP methodologies, the benchmarks, and the analysis of the results obtained from our experiments.

7.1 Environment Setup

The GPU used in our experiments was the GeForce GTX TITAN. This processor has 2,688 CUDA cores (at 837 MHz) and 6 GB of RAM (no ECC) with a memory bandwidth of 288.4 GB/s through a 384-bit data bus. The GTX TITAN GPU is based on the Nvidia Kepler architecture, and its theoretical peak performance is characterized by the use of the fused multiply-add (FMA) operations. The GTX TITAN can achieve single precision theoretical peak performance of 4.5 TFLOPs.

GMGP creates the individuals on CPU using a single-threaded code running on a single core of an Intel Xeon CPU X5690 processor, with 32 KB of L1 data cache, 1.5 M of L2 cache, 12 MB of L3 cache, and 24 GB of RAM, running at 3.46 GHz.

The GP methodologies were implemented in C, CUDA 5.5, and PTX 3.2. The compilers used were gcc 4.4.7, nvcc release 5.5, V5.5.0, and ptxas release 5.5, V5.5.0. We had to be careful in setting the compiler optimization level. It is common for the programmer to use a more advanced optimization level to produce a more optimized and faster code. However, the compilation time is a bottleneck for the GP methodologies that require individuals to be compiled. The code generated by the -O2, -O3, and -O4 optimization levels is more optimized and executes faster, but more time is spent in the compilation process. Experiments were performed to determine the best optimization level. These experiments indicated that the lowest optimization level, -O0, provided the best results. There were millions of individuals to be compiled, and each individual was executed only once. Accelerating the execution phase was not sufficient to compensate for the time spent optimizing the code during the compilation phase.

We used five widely used GP benchmarks: two symbolic regression problems, *Mexican Hat* and *Salutowicz*; one time-series forecasting problem, *Mackey-Glass*; one image processing problem, *Sobel filter*; and one Boolean regression problem, *20-bit Multiplexer*. The first four benchmarks were used to evaluate the single precision floating point instructions, whereas the last benchmark was used to evaluate the Boolean instructions. The *Mackey-Glass*, *Boolean Multiplexer*, and *Sobel filter* benchmarks were also used in previous works on GP accelerated by GPUs (Robilliard et al., 2009; Langdon and Banzhaf, 2008c; Langdon, 2010b; Harding and Banzhaf, 2008, 2009). Nevertheless, it is not possible to perform a direct comparison, as they used a different GP model (tree-based GP) and different hardware.

Each result in the experiments was obtained by repeating the experiment 10 times and averaging the timing results. The standard deviations of the times obtained for all the data sets were less than 5% of the average execution times. We present our timing results in both seconds and GP operations per second (GPops), which has been widely used in previous GP works. Although the focus of the paper is on the actual execution speeds of the GP evaluation, we briefly discuss the quality of the results produced by GMGP and the other methodologies studied.

We used 256 threads per block in our experiments. The block grid is two-dimensional and depends on the number of individuals and the number of fitness cases. For an experiment with *(number_fitness cases, number_individuals)* the grid is *(number_fitness cases/256, number_individuals)*.

7.2 GP Implementations

To put the GMGP results in perspective, we compare the performance of GMGP with the other GP methodologies for GPUs: *compilation*, *pseudo-assembly*, and *interpretation*. However, the GP methodologies for GPUs taken from the literature are not based on LGP or quantum-inspired algorithms. For this reason, we had to implement an LGP and quantum-inspired approach corresponding to each methodology to make them directly comparable with GMGP. Nevertheless, these implementations are based on the algorithms described in the literature.

The *compilation* approach is based on the work by Harding and Banzhaf (2009) and is called **Compiler** here. The *pseudo-assembly* approach is based on our previous work (Cupertino et al., 2011), and is called **Pseudo-Assembly** here. The *interpretation* approach is based on the work by Langdon and Banzhaf (2008a) and is called **Interpreter** here.

The Compiler and Pseudo-Assembly methodologies use a similar program assembly to the GMGP methodology. The individuals are created by the CPU and sent to the GPU to be computed. The main difference is the assembly of the body of the programs. In Compiler, the bodies are created using CUDA language instructions. When the population is complete, it is compiled using the *nvcc* compiler to generate the GPU binary code. In Pseudo-Assembly, the bodies are created using the PTX pseudo-assembly language instructions. When the population is complete, the code can be compiled with *ptxas* or the *cuModuleLoad* C function provided by Nvidia, both of which generate GPU binary code. The Pseudo-Assembly methodology reduces the compilation overhead using the JIT compilation.

In the Interpreter methodology, the interpreter was written in the PTX language, rather than in RapidMind, as proposed by Langdon and Banzhaf (2008a). The interpreter is automatically built once, at the beginning of the GP evolution, and is reused to evaluate all individuals. Algorithm 1 presents a high-level description of the interpreter process. As the pseudo-assembly language does not have a switch-case statement, we used a combination of the instruction `setp.eq.s32` (comparisons) and `bra` (branches) to obtain the same functionality. These comparisons and branches represent one of the weaknesses of the Interpreter methodology. The interpreter must execute more instructions than the actual GP operations. For each GP instruction, we have at least one comparison, to identify the GP operation, and one jump to the beginning of the loop. In addition, comparisons can be made to identify the instruction arguments.

The GP methodologies implemented employ an equivalent function set and use the same number of registers. In QILGP (Dias and Pacheco, 2013), the function set has an atomic exchange instruction (`FXCH ST(i)`) that the GPU does not have. To maintain the function set compatibility with QILGP in the experiments, we created the exchange operation in the GPU using three move operations. An exchange between Ri and $R0$ uses an intermediary register $R8$ and becomes $R8 = R0$; $R0 = Ri$; $Ri = R8$, as shown in Table 2.

```
 1: TBX← X dimension of the Thread Block identification
 2: TBY← Y dimension of the Thread Block identification
 3: INDIV← individual number (TBY)
 4: N← program length (INDIV)
 5: THREAD← GPU Thread identification
 6: X0← input variable 1 (THREAD + TBX * Number of threads in a block)
 7: X1← input variable 2 (THREAD + TBX * Number of threads in a block)
 8: for k ← 1 to N do
 9:    INSTRUCT← instruction number (k) (INDIV)
10:    ARG← argument number (k) (INDIV)
11:    switch (INSTRUCT)
12:    case 0:
13:       no operation
14:    case 1:
15:       switch (ARG)                          % Description: R(0) ← R(0) + X(j)
16:       case 0:
17:          add.f32 R0, R0, X0
18:       case 1:
19:          add.f32 R0, R0, X1
20:       → Here, we have more similar cases for all inputs and constant registers (X).
21:       end switch
22:    case 2:
23:       switch (ARG)                          % Description: R(0) ← R(0) + R(i)
24:       case 0:
25:          add.f32 R0, R0, R0
26:       case 1:
27:          add.f32 R0, R0, R1
28:       → Here, we have more similar cases for all eight auxiliary FPU registers (Ri).
29:       end switch
30:    case 3:
31:       switch (ARG)                          % Description: R(i) ← R(i) + R(0)
32:       case 0:
33:          add.f32 Ri, R0, R0
34:       case 1:
35:          add.f32 Ri, R1, R0
36:       → Here, we have more similar cases for all eight auxiliary FPU registers (Ri).
37:       end switch
38:    → Here, we have more similar cases for all other instructions, such as subtraction, multiplication,
       division, data transfer, trigonometric, and arithmetic operations.
39:    default:
40:       exit
41:    end switch
42:    → Write result back to global memory.
43: end for
```

Algorithm 1: Pseudo-code for the GP interpreter for a GPU based on quantum-inspired LGP.

7.3 Symbolic Regression Benchmarks

Symbolic regression is a typical problem used to assess GP performance. We used two well-known benchmarks: the *Mexican Hat* and *Salutowicz*. These benchmarks allow us to evaluate GMGP over different fitness case sizes.

The *Mexican Hat* benchmark (Brameier and Banzhaf, 2007) is represented by a two-dimensional function given by Equation (8):

$$f(x,y) = \left(1 - \frac{x^2}{4} - \frac{y^2}{4}\right) \times e^{(-x^2-y^2)/8}. \tag{8}$$

The *Salutowicz* benchmark (Vladislavleva et al., 2009) is represented by Equation (9). We used the two-dimensional version of this benchmark.

$$f(x,y) = (y-5) \times e^{-x} \times x^3 \times cos\,(x) \times sin\,(x) \times \left[cos\,(x) \times sin\,(x)^2 - 1\right]. \tag{9}$$

For the *Mexican Hat* benchmark, the x and y variables are uniformly sampled in the range [−4,4]. For the *Salutowicz* benchmark, they are uniformly sampled in the range [0,10]. This sampling generates the training, validation, and testing data sets. The number of subdivisions of each variable can be 16, 32, 64, 128, 256, and 512, which is called the number of samples, N. At each time, both variables use the same value of N, producing a grid. When $N = 16$, there is a 16×16 grid, which represents 256 fitness cases. Accordingly, the number of fitness cases varies in the set $S = \{256, 1024, 4096, 16K, 64K, 256K\}$.

These two benchmarks represent two different surfaces, and GP has the task of reconstructing these surfaces from a given set of points. The fitness value of an individual is its mean absolute error (MAE) over the training cases, as given by Equation (10):

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |t_i - V[0]_i|, \tag{10}$$

where t_i is the target value for the *ith* case and $V[0]_i$ is the individual output value for the same case.

7.3.1 Parameter Settings

Table 5 presents the parameters used when executing the *Mexican Hat* and *Salutowicz* benchmarks. We used a small population size, which is a typical characteristic of QEAs. The evolution status of QEAs is represented by a probability distribution, and there is no need to include many individuals. The superposition of states provides a good global search ability due to the diversity provided by the probabilistic representation.

7.3.2 Preliminary Experiments for the Compiler Methodology

Table 6 presents the execution time breakdown of all GPU methodologies for the *Mexican Hat* benchmark when the fitness case is 16K. The execution time is broken down into the following categories: `nvcc` represents the time spent with the *nvcc* compiler to generate the PTX code from the CUDA source code; `upload` represents the time spent compiling the PTX code to the GPU binary code (in our methodology, `upload` means the time spent loading

Parameter	Settings	
	Mexican Hat	Salutowicz
Number of generations	400,000	400,000
Population size	36	36
NOP initial probability ($\alpha_{0,0}$)	0.9	0.9
Step size (s)	0.0003	0.002
Maximum program length	128	128
Function set	Table 2	Table 2
Set of constants	{1,2,3,4,5,6,7,8,9}	{1,2,3,4,5,6,7,8,9}

Table 5: Parameter settings for the Mexican Hat and Salutowicz benchmarks. The values of number of generations, population size, initial probability of NOP, and step size were obtained from previous experiments.

Methodology	Total	nvcc	upload	evaluation	interpret	download	CPU
GMGP	292.6	–	73.2	76.9	–	5.13	137.2
Interpreter	636.8	–	3.14	–	542.4	4.35	86.8
Pseudo-Assembly	40,777	–	40,414	118.8	–	6.13	238.8
Compiler	242,186.7	135,027.5	106,458	283.6	–	6.74	410.9

Table 6: Execution time breakdown of all GPU methodologies (in seconds). The table presents the times for: `Total`, the total execution; `nvcc`, the compilation in the *nvcc* compiler; `upload`, the compilation of the PTX code (Compiler and Pseudo-Assembly), or loading the GPU binaries to the GPU memory (GMGP), or transferring the tokens through the PCIe bus (Interpreter); `evaluation`, the computation of the fitness cases; `interpret`, the interpretation; `download`, the copy of the fitness result from GPU to the CPU; and `CPU`, the GP methodology is executed on the CPU.

the GPU binaries to the graphic card before execution); in the interpreter methodology, `upload` is the time necessary to transfer the tokens through the PCIe bus; `evaluation` represents the time spent computing the fitness cases; `interpret` is the interpretation time for Interpreter; `download` is the time spent in copying the fitness result from GPU to the CPU; and `CPU` represents the remainder of the execution time, including the time necessary to execute the GP methodology on the CPU.

As can be observed in Table 6, the Compiler methodology is the only one that spends time on the *nvcc* compiler. The time spent on the *nvcc* compiler is enormous when compared to all other times, and Compiler becomes three orders of magnitude slower than GMGP and Interpreter. Although some previous works have reported results for the Compiler methodology for GP in GPUs (Harding and Banzhaf, 2007; Chitty, 2007; Harding and Banzhaf, 2009; Langdon and Harman, 2010), they are not comparable with our results. Harding and

Banzhaf (2007) and Chitty (2007) did not use CUDA and could therefore avoid the *nvcc* overhead. Harding and Banzhaf (2009) used CUDA but handled the compilation overhead by using a cluster to compile the population. Langdon and Harman (2010) also used CUDA, but the total compilation time for our experiment is greater than their compilation time for two reasons. First, the small population size of a quantum-inspired approach requires more compiler calls. Second, the total number of individuals we are evaluating (number of generations × population size) is at least one order of magnitude greater than in their experiments.

Because the other methodologies solved the same problem considerably faster, we discarded the Compiler methodology for the remaining experiments.

The `download` time is almost the same for all implementations because the same data set was used in all approaches. Accordingly, the results to be copied through the PCIe bus are the same. The `CPU` time for Interpreter is slightly smaller than for GMGP, Compiler, and Pseudo-Assembly because Interpreter does not have to assemble the individuals in the CPU before transferring to the GPU. Instead, the tokens are copied directly. The `evaluation` time is almost the same for Compiler and Pseudo-Assembly, but GMGP presents a slightly smaller `evaluation` time because the header and footer are optimized. The `interpret` time is approximately one order of magnitude slower than the GMGP `evaluation` time because it has to perform many additional instructions, such as comparisons and jumps. The `upload` time for GMGP is approximately three orders of magnitude faster than the `upload` time for Compiler and Pseudo-Assembly because GMGP directly assembles the GPU binaries without calling the PTX compiler. The time necessary to transfer the tokens through the PCIe bus in the Interpreter methodology is smaller than the time necessary to load the GPU binary code in the GMGP.

7.3.3 Performance Analysis

We compare the execution times of the methodologies as the number of fitness cases varies in the set: $S = \{256, 1024, 4096, 16K, 64K, 256K\}$. The total execution times of the *Mexican Hat* and *Salutowicz* benchmarks for the Pseudo-Assembly, Interpreter, and GMGP methodologies are presented in Figure 5. The curves are plotted in log-scale. The Pseudo-Assembly methodology execution time remains almost constant as the problem size increases in both cases studied because Pseudo-Assembly spends most of the time compiling the individual population code, and the compilation time does not depend on the problem size. The total execution times of the Interpreter and GMGP methodologies increase almost linearly as the number of fitness cases increases from 256 to 256K. For the largest data set, 256K, the Pseudo-Assembly execution time approaches the execution time of Interpreter. However, the Pseudo-Assembly methodology performs much worse than the other methodologies when only a few fitness cases are considered.

In Table 7, we present the performance of the three methodologies for a 256K data set, using the GP operations per second (GPops) metric, which is widely employed in the GP literature. Considering the total evolution, GMGP performs 2.29e+014 GP operations on 1.17e+003 seconds, obtaining 194.4 billion GPops for *Mexican Hat*. Similarly, GMGP obtained 200.5 billion GPops for *Salutowicz*. The Interpreter methodology took 26.6 billion GPops for *Mexican Hat* and 27.5 billion GPops for *Salutowicz*. The Pseudo-Assembly

Benchmark	Methodology	GP evolution (GPops)	Best Individual (GPops)
Mexican Hat	GMGP	194.4 billion	245.5 billion
	Interpreter	26.6 billion	29.9 billion
	Pseudo-Assembly	5.3 billion	161.0 billion
Salutowicz	GMGP	200.5 billion	240.2 billion
	Interpreter	27.5 billion	27.0 billion
	Pseudo-Assembly	4.9 billion	158.4 billion

Table 7: Performance of GMGP, Interpreter, and Pseudo-Assembly for *Mexican Hat* and *Salutowicz* in GPops. The table presents the results for the overall evolution, including the time spent in the GPU and CPU, and the results for the GPU computation of the best individual after the evolution is complete.

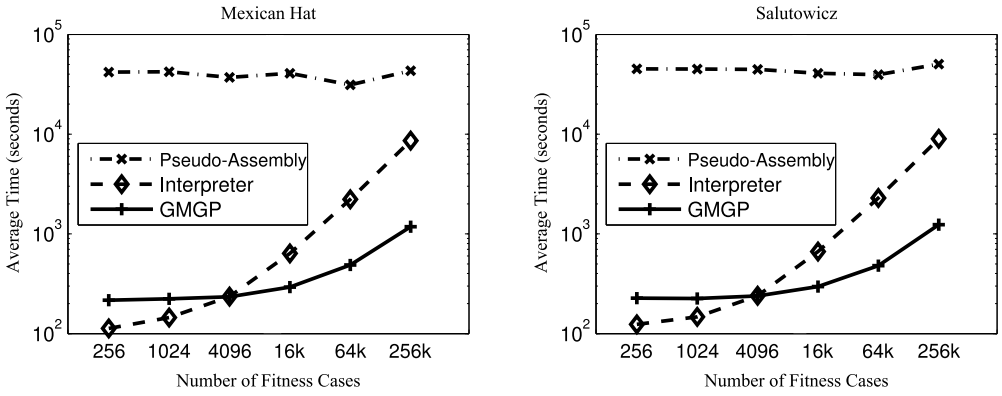


Figure 5: Execution time (in seconds) of Pseudo-Assembly, Interpreter, and GMGP methodologies for the *Mexican Hat* and *Salutowicz* benchmarks with an increasing number of fitness cases.

methodology had the smallest values, 5.3 billion GPops for *Mexican Hat* and 4.9 billion GPops for *Salutowicz*. Table 7 also presents the GPops for the evaluation in the GPU of the best individual found after the evolution is completed. The best individual GPops results are greater than the GP evolution results because the evaluation of the best individual takes considerably less time than the whole GP evolution. In addition, the GP evolution includes the overheads of creating the individuals and transferring the data to/from the GPU. For Pseudo-Assembly, the evaluation of the best individual does not consider the compilation overhead, and the GPops value obtained for the best individual is similar to that obtained by GMGP.

Figure 6 presents the speedups obtained with the Interpreter and GMGP methodologies compared to the Pseudo-Assembly methodology for the *Mexican Hat* and *Salutowicz* benchmarks. For the two benchmarks, the smallest data set generated the greatest speedups. For

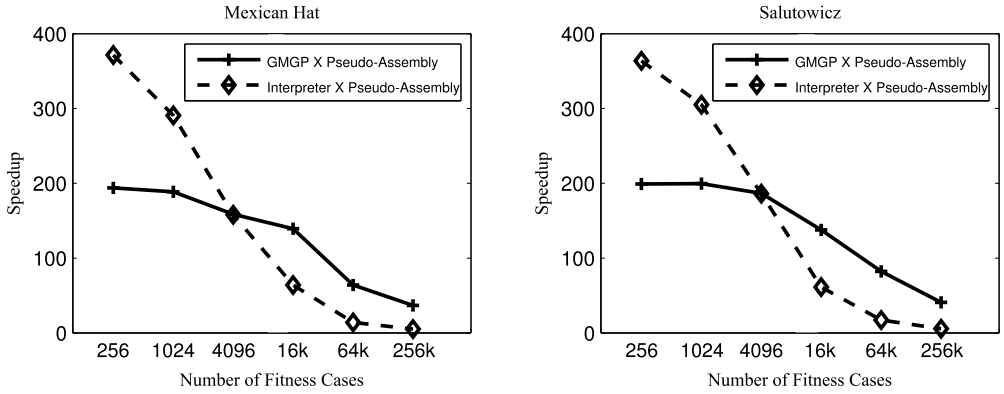


Figure 6: Speedup of Interpreter and GMGP compared to Pseudo-Assembly for the *Mexican Hat* and *Salutowicz* benchmarks with an increasing number of fitness cases.

Mexican Hat, Interpreter runs 371 times faster than Pseudo-Assembly, whereas GMGP runs 193 times faster than Pseudo-Assembly. The gains are similar for *Salutowicz*: Interpreter runs 363 times faster than Pseudo-Assembly, and GMGP runs 199 times faster than Pseudo-Assembly. As the problem size increases, the speedups compared to Pseudo-Assembly become smaller for both benchmarks. We will compare only Interpreter and GMGP in the remainder of this analysis.

Figure 7 presents the speedup obtained with GMGP compared to Interpreter for *Mexican Hat* and *Salutowicz*. GMGP performs better for larger data sets for both benchmarks. For the small data sets, in GMGP, the number of fitness cases used is not sufficient to compensate for the overhead of uploading the individuals, and the Interpreter methodology is faster. GMGP outperforms Interpreter for fitness case sizes exceeding 4,096. GMGP is 7.3 times faster than Interpreter for *Mexican Hat* and a fitness case size of 256K. Similar results were obtained for *Salutowicz*. As expected, GMGP is promising for applications with large data sets.

To explain why GMGP outperforms Interpreter for large data sets, we analyze the execution time breakdown for each approach in detail. Figures 8 and 9 present the execution breakdown of GMGP and Interpreter for the *Mexican Hat* and *Salutowicz* benchmarks with an increasing number of fitness cases. The execution time was broken into the same components as described in Table 6.

A comparison of GMGP's `upload` time from Figure 8 with Interpreter's `upload` time from Figure 9 indicates that it is more costly to load the GPU binary to the graphics card than to transfer the tokens through the PCIe bus. However, these times remain constant as the problem size increases. The `download` times for GMGP and Interpreter are almost the same, but both times increase with increasing problem size. This result is expected, as the two approaches use exactly the same data set, and the computations produce the same number of results to be copied through the PCIe bus. The result of each thread execution is one float value. The results of the threads in one block are reduced to one result in the global

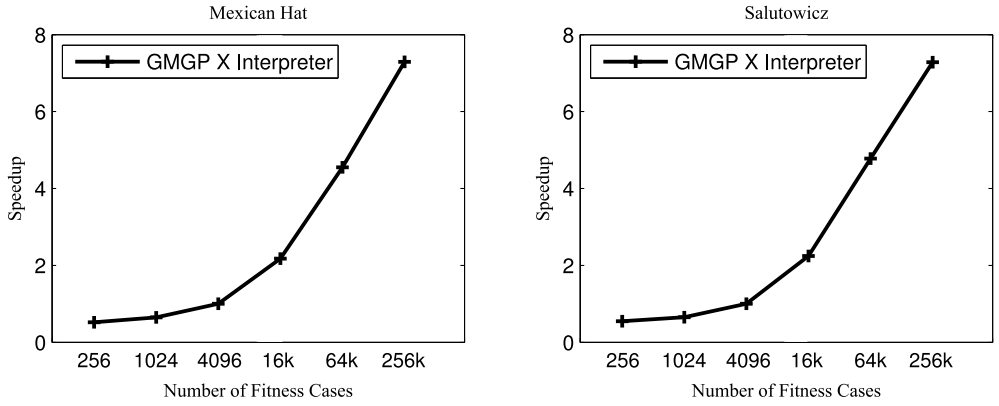


Figure 7: Speedup of GMGP compared to Interpreter for the *Mexican Hat* and *Salutowicz* benchmarks with an increasing number of fitness cases.

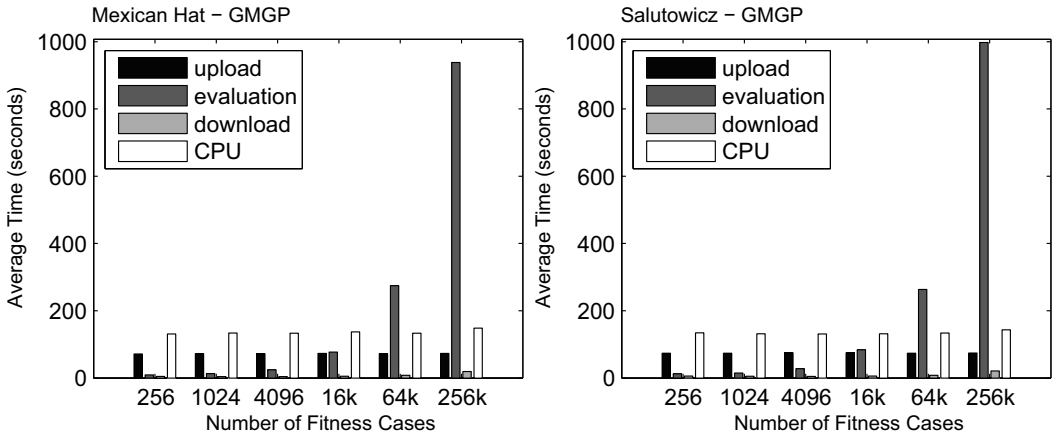


Figure 8: Execution time breakdown of GMGP. The graph presents the time broken down as follows: upload, the time spent loading the GPU binaries to the GPU memory; evaluation, the time spent computing the fitness cases; download, the time spent copying the fitness result from the GPU to the CPU; and CPU, the time during which the GP methodology is executed on the CPU.

memory. Then, the block results are reduced to one value for each individual in the CPU. The number of results transferred depends on the number of blocks used to compute all of the fitness cases. The `CPU` overhead has a similar behavior because the time spent running the GP methodology on the CPU is expected to be the same for GMGP and Interpreter, as the parallelized portion of the code is the evaluation function. We can compare the

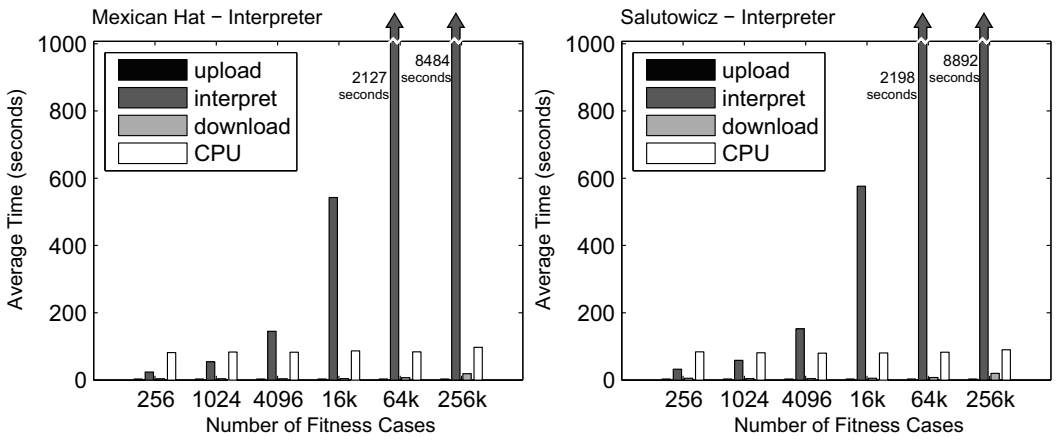


Figure 9: Execution time breakdown of Interpreter. The graph presents the time broken into: upload, the time necessary to transfer the tokens through the PCIe bus; interpret, the interpretation time; download, the time spent copying the fitness result from the GPU to the CPU; and CPU, the time during which the GP methodology is executed on the CPU.

evaluation function times for GMGP and Interpreter by comparing the `evaluation` time of Figure 8 with the `interpret` time of Figure 9. For small data sets, the `evaluation` time of GMGP is smaller than the `interpret` time of Interpreter, but the difference is small. However, as the problem size increases, the `interpret` time increases significantly because the Interpreter methodology must execute an excessive amount of additional instructions, such as comparisons and branches. For GMGP, the `evaluation` time increases slightly because it executes only the necessary GP instructions. Thus, the total time difference between GMGP and Interpreter increases for larger data sets.

7.3.4 Quality of Results

To compare the quality of the results of the Compiler, Pseudo-Assembly, Interpreter, and GMGP methodologies on the GPU, we used the same random seed at the beginning of the first experiment of each approach. We compared the intermediate and final results. All GPU approaches produced identical results, comparing all available precision digits. The only difference among them was the execution time.

In Table 8, we analyze the results for 10 different executions of Compiler, Pseudo-Assembly, Interpreter, and GMGP. Table 8 presents the best individuals' average and standard deviation (σ) for the training, validation, and testing data sets for the *Mexican Hat* and *Salutowicz* benchmarks considering 16K fitness cases. Because each experiment was repeated 10 times, the standard deviations of all cases are relatively low for the number of executions used.

Benchmark	Methodology	Training		Validation		Test	
		Average	σ	Average	σ	Average	σ
Mexican Hat	GMGP	0.046	0.007	0.048	0.008	0.053	0.008
	Interpreter	0.046	0.007	0.048	0.008	0.053	0.008
	Pseudo-Assembly	0.046	0.007	0.048	0.008	0.053	0.008
	Compiler	0.046	0.007	0.048	0.008	0.053	0.008
Salutowicz	GMGP	0.17	0.10	0.19	0.12	0.15	0.08
	Interpreter	0.17	0.10	0.19	0.12	0.15	0.08
	Pseudo-Assembly	0.17	0.10	0.19	0.12	0.15	0.08
	Compiler	0.17	0.10	0.19	0.12	0.15	0.08

Table 8: Mean Absolute Errors (MAEs) in GPU evolution for the *Mexican Hat* and *Salutowicz* benchmarks. The table presents the best individuals' average and standard deviation (σ) for the training, validation, and testing data sets for 16K fitness cases, with a precision of 10^{-3}.

7.4 Mackey-Glass Benchmark

The *Mackey-Glass* benchmark (Jang and Sun, 1993) is a chaotic time-series prediction benchmark, and the *Mackey-Glass* chaotic system is given by the non-linear time delay differential Equation (11).

$$\frac{dx(t)}{dt} = \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t) \tag{11}$$

The *Mackey-Glass* system has been used as a GP benchmark in various works (Langdon and Banzhaf, 2008b,c). In our experiments, the time series consists of 1,200 data points, and GP has the task of predicting the next value when historical data are provided. The GP inputs are eight earlier values from the series, at 1, 2, 4, 8, 16, 32, 64, and 128 time steps ago.

7.4.1 Parameter Settings

The parameters used for the GP evolution in the *Mackey-Glass* benchmark are presented in Table 9. We used a small population size and a large number of generations, as previously explained. The number of generations was defined according to the number of individuals proposed by Langdon and Banzhaf (2008c).

7.4.2 Performance Analysis

We analyze the performance of GMGP for the *Mackey-Glass* benchmark using the GPops metric. Table 10 presents the number of GPops obtained by GMGP. We present the GPops for the GP evolution in the GPU considering the operations spent in executing the evaluation function for all individuals and counting all non-`NOP` operations. GMGP obtained 77.7 billion GPops. When we consider the GP evaluation combined with the load of the

Parameter	Settings
Number of generations	512,000
Population size	20
NOP initial probability ($\alpha_{0,0}$)	0.9
Step size (s)	0.004
Maximum program length	128
Function set	Table 2
Set of constants	{0, 0.01, 0.02, ..., 1.27}

Table 9: Parameter settings for the *Mackey-Glass* benchmark. The number of individuals (number of population x number of generations) was defined according to the literature. The initial probability of NOP and step size were obtained in previous experiments.

	GPops
GP evolution	77.7 billion
+ loading data	8.85 billion
+ results transfer	8.4 billion
Total computation	3.59 billion
Best individual	8.6 billion

Table 10: Results of GMGP running the *Mackey-Glass* benchmark in GPops. The table presents the number of GPops spent in the GP evolution in the GPU, progressively including the overhead of loading the individuals code into GPU and transferring the results back to the CPU. At the end, we provide the results for the entire computation, including the overhead of CPU computation, and the results for the execution of the best individual.

individual code into the GPU memory, GMGP obtained 8.85 billion GPops. The load of data into the GPU memory does not include any GP operation and requires a substantial time in the evolution process. The load time is fixed regardless of the size of the data set. The idea is to amortize this cost by the faster execution of a larger data set. However, the *Mackey-Glass* benchmark has a small number of fitness cases.

For the measures that consider the transfer of the results to the CPU memory, the GPops value decreased to 8.4 billion. When the entire computation is considered, including the overhead of the CPU computation, GMGP achieved 3.59 billion GPops. At the end of the evolution, the best individual was executed, and the performance of the best individual execution was 8.6 billion GPops.

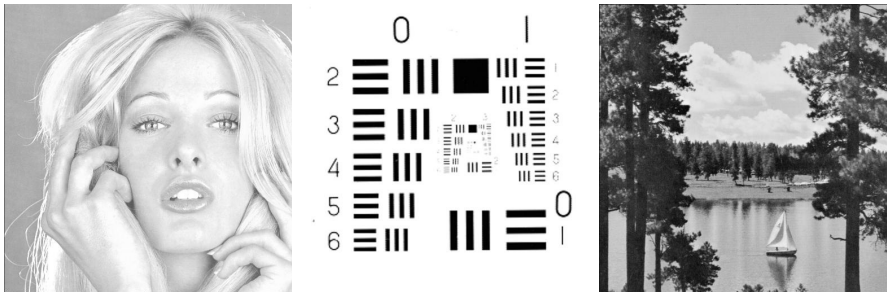


Figure 10: The three gray-scale images used for training. The image resolutions are 512 × 512 pixels.

7.4.3 Quality of the Results

The quality of the results produced by GMGP was analyzed using 10 GP executions. We computed the RMS error and standard deviation. The average error was 0.0077, and the standard deviation was 0.0021. The error is lower than the errors presented in the literature due to the difference in the GP models used. The results presented in the literature used a tree-based GP with a tree size limited to 15 and depth limited to 4. In contrast, GMGP can evolve individuals with at most 128 linear instructions. Accordingly, it was possible to find an individual that better addressed this benchmark problem.

7.5 Sobel filter

The *Sobel filter* is a widely used edge detection filter. Edges characterize boundaries and are therefore considered crucial in image processing. The detection of edges can assist in image segmentation, data compression, and image reconstruction. The Sobel operator calculates the approximate image gradient of each pixel by convolving the image with a pair of 3×3 filters. These filters estimate the gradients in the horizontal (x) and vertical (y) directions, and the magnitude of the gradient is the sum of these gradients. All edges in the original image are greatly enhanced in the resulting image, and the slowly varying contrast is suppressed.

The evolution of an image filter uses a reverse-engineering approach, where the problem is to find the mapping between the original image and resulting image after the filter is applied (Harding and Banzhaf, 2008, 2009). The GP task is to discover the operations that transformed the input image into the filtered image. In our experiments, we used six 512×512 images taken from the USC-SIPI image repository (Weber, 1997). The gray-scale versions of all 6 images and the resulting images after the *Sobel filter* were computed using the GIMP image processing tool (GIMP, 2008). Figure 10 presents the three gray-scale images used for training. Figure 11 shows the two images used for validation. Figure 12 shows, for the same image, the original image in gray scale, the resulting image after the *Sobel filter* is applied by the GIMP tool, and the output image produced by the GMGP evolved filter.

Figure 11: The two gray-scale images used for validation. The image resolutions are 512 × 512 pixels.

Figure 12: Results of evolving the filter for one test image. The leftmost image is the original gray-scale test image. The center image is the output image produced by applying the GIMP Sobel filter. The rightmost image is the output image produced by the GMGP evolved filter.

7.5.1 Parameter Settings

The parameters used for the GP evolution of the *Sobel filter* are presented in Table 11. The population size also employs a low number of individuals for the reasons explained before. The number of generations, NOP initial probability, step size, and maximum program length were obtained from previous experiments.

7.5.2 Performance Analysis

The performance of the *Sobel filter* in GPops is presented in Table 12. Considering only the GPU evaluation of all non-`NOP` instructions, GMGP achieved 287.3 billion GPops. When the overhead of uploading the GPU binaries is included, the GPops are reduced to 274.2 billion. The reduction in GPops was less pronounced because this problem has a larger data set that compensates for the initial overhead of loading the program. When we include the overhead

Parameter	Settings
Number of generations	400,000
Population size	20
NOP initial probability ($\alpha_{0,0}$)	0.9
Step size (s)	0.001
Maximum program length	128
Function set	Table 2
Set of constants	{1,2,3,4,5,6,7,8,9}

Table 11: Parameter settings for the *Sobel filter*. The values of the number of generations, population size, initial probability of NOP, and step size were obtained in previous experiments.

	GPops
GP evolution	287.3 billion
+ loading data	274.2 billion
+ results transfer	268.6 billion
Total computation	249.9 billion
Best individual	295.8 billion

Table 12: Results of GMGP running the *Sobel filter* in GPops. The table presents the number of GPops spent in the GP evolution in the GPU, progressively including the overheads of loading the individual code into the GPU and transferring the results back to the CPU. At the end, we present the results for the entire computation, including the overhead of CPU computation, and the results for the execution of the best individual.

of transferring the results back to the CPU through the PCIe bus, GMGP obtained 268.6 billion GPops. When the entire computation is considered, including the overhead of the CPU computation during the evolution, GMGP obtained 249.9 billion GPops. After the evolution, the best individual was executed on the GPU, and we calculated a performance of 295.8 billion GPops for the best individual.

7.5.3 Quality of the Results

The quality of the results produced by GMGP for the *Sobel filter* was analyzed with 10 GP runs. We computed the MAE and standard deviation. Table 13 presents both the MAEs and standard deviations for the training, validation, and testing data sets. The errors are low compared to those presented in literature because our GP parameters were set to provide a better-quality evolved filter. The quality of the *Sobel filter* evolved by GMGP can also be assessed visually. The image presented at the right of Figure 12 was produced by the

Training		Validation		Test	
Average	σ	Average	σ	Average	σ
2.11	0.61	2.21	0.64	2.03	0.599

Table 13: MAEs in GMGP evolution for the *Sobel filter*. The table presents the average and standard deviation (σ) of the best individual for the training, validation, and testing data sets.

best individual of GMGP applied to the test image. This image can be visually compared to the image in the center of Figure 12, which was obtained using the *Sobel filter* of GIMP. A visual comparison of these two images indicates that the evolved filter produced an image with more prominent horizontal edges without significantly increasing the noise.

7.6 20-bit Boolean Multiplexer

The Boolean instructions of GMGP were evaluated using the *20-bit Boolean Multiplexer* benchmark (Langdon, 2010b, 2011). In the *20-bit Boolean Multiplexer* benchmark, there are 1,048,576 possible combinations of 20 arguments of a 20-bit Multiplexer. In our experiments, we used 1,048,576 fitness cases to evaluate all of the individuals, which is possible because GMGP evaluates each individual rapidly. This experiment is the first time this benchmark has been solved in this manner, using all fitness cases. The bit-level parallelism was exploited by performing bitwise operations over a 32-bit integer that packs 32 Boolean fitness cases.

7.6.1 Parameter Settings

The parameter settings used for the *20-bit Boolean Multiplexer* benchmark are presented in Table 14. More individuals were used in the population than in the previous benchmark experiments reported in this paper. This problem addresses more input variables and a larger data set. The number of generations was computed to produce a total number of individuals similar to the numbers presented in the literature. However, the zero error solution was found before the maximum number of generations was reached for all 10 GP executions. The maximum program length was obtained by verifying the minimum length needed to solve this problem benchmark.

7.6.2 Performance Analysis

Table 15 presents the number of GPops obtained by GMGP for the GP evolution in the GPU (execution of the evaluation of all individuals considering the non-`NOP` operations); the GP evolution including the loading of the individual code into the GPU memory; the GP evolution, including the loading of the individuals and the transfer of the results to the CPU memory; the total computation, including the CPU computation; and the best individual computation.

Table 15 illustrates that GMGP obtained 5.88 trillion GPops when evaluating the individuals. When the load of the individuals is considered, a value of 5.24 trillion GPops was obtained. This benchmark has a large data set. The amount of computation is sufficient to

Parameter	Settings
Number of generations	First Solution or 14,000,000
Population size	40
NOP initial probability ($\alpha_{0,0}$)	0.9
Step size (s)	0.004
Maximum program length	512
Function set	Table 3
Set of constants	–

Table 14: Parameter settings for the *20-bit Boolean Multiplexer*. The values of the number of generations, population size, initial probability of NOP, and maximum program length were obtained in previous experiments, where we varied the values until the problem was solved.

	GPops
GP evolution	5.88 trillion
+ loading data	5.24 trillion
+ results transfer	5.19 trillion
Total computation	2.74 trillion
Best individual	4.87 trillion

Table 15: Results of GMGP running the *20-bit Boolean Multiplexer* benchmark in GPops. The table presents the number of GPops spent in the GP evolution in the GPU, progressively including the overhead of loading the individual code into GPU and transferring the results back to the CPU. At the end, we present the results for the entire computation, including the overhead of CPU computation, and the results for the execution of the best individual.

amortize the load time. Thus, the total number of GPops is not degraded with the inclusion of the load of individuals. When the results transfer is included, the results remain almost the same, and GMGP achieves 5.19 trillion GPops. When the CPU overhead is considered, the performance is reduced to 2.74 trillion GPops. This result suggests that porting the whole GP evolution algorithm to run in the GPU (not only the evaluation function), could significantly improve the overall performance. The execution of the GMGP's best individual achieved 4.87 trillion GPops.

Experiment	Generation	Total Number of Individuals
1	2,413,505	96,540,200
2	1,246,979	49,879,160
3	3,238,394	129,535,760
4	7,802,509	312,100,360
5	8,892,873	355,714,920
6	10,737,990	429,519,600
7	5,255,728	210,229,120
8	2,576,655	103,066,200
9	5,469,381	218,775,240
10	3,395,730	135,829,200

Table 16: Generation at which GMGP solved the *20-bit Boolean Multiplexer* and the total number of individuals used in the evolution. The population size is 40 individuals.

7.6.3 Quality of the Results

GMGP was able find the zero solution for the *20-bit Boolean Multiplexer* benchmark before the maximum number of generations was reached for all 10 GP executions. Table 16 presents the number of generations and total number of individuals needed to find this solution.

8. Discussions

It is difficult to compare our quantum-inspired LGP timings to the timings of the tree-based implementations of GP in GPU proposed in the literature. They used different individual representations and different evolutionary algorithms. However, we can compare the GPops results. For the *Mackey-Glass* benchmark, on the GTX TITAN, we obtained up to 3.59 billion GPops when considering the entire evaluation (GPU and CPU) and 77.7 billion GPops when considering only the GPU processing. Langdon and Banzhaf (2008c) obtained 895 million GPops for this benchmark. However, we used a larger individual than Langdon and Banzhaf (2008c) to achieve a more accurate prediction result (smaller RMS error). We obtained up to 249.9 billion GPops considering the whole evaluation (GPU and CPU) and 287.3 billion GPops considering only the GPU processing for the *Sobel filter* benchmark. The *Sobel filter* was also evolved, along with other filters, by Harding and Banzhaf (2008).They obtained an average of 145 million GPops and a peak performance of 324 million GPops. Harding and Banzhaf (2009) attained on average 4.21 billion GPops when evolving the same type of filter. They used Cartesian GP and a cluster of 16 workstations to compile the code. For the *20-bit Boolean Multiplexer* benchmark, we obtained up to 2.74 trillion GPops considering the entire evaluation and 5.88 trillion GPops considering only the GPU processing. Langdon (2010b) obtained up to 254 billion GPops in the entire evaluation process (CPU and GPU) for a 37-bit Boolean multiplexer. The literature provides other results for different benchmarks. Recently, Langdon (2010a) obtained 8.5 billion GPops for a bioinformatics data mining problem.

Despite the highly data-parallel nature of the GP problems that we considered, we could achieve 336.3 GFLOPs of execution in the GTX TITAN, whose peak performance is 4,500 GFLOPs, running the *Sobel filter* benchmark. The peak performance of the GPU is measured using the FMA instruction, which is not present in our function set. Furthermore, it is difficult to reach the peak single precision performance even for embarrassingly parallel applications, such as SGEMM (Lai and Seznec, 2013). The main obstacles for GMGP in achieving the peak performance are: (i) it includes more complicated floating-point operations like divisions, sine, cosine, and square root, that take several cycles to execute; (ii) it includes a reduction operation that requires synchronization; (iii) the small population size makes the overhead of uploading the individuals to the GPU memory more significant.

9. Conclusions

In this work, we proposed a new methodology to parallelize the evaluation process on the GPU called **GMGP**. Our methodology is inspired by quantum computing and includes the principles of the quantum bit and the superposition of states, which increases the diversity of a quantum population. In addition, GMGP is the first methodology to generate individuals using the GPU machine code instead of compiling or interpreting them. We eliminate the compilation time overhead without including the parsing of the code and divergence required for the interpretation. The parallelism is exploited at two levels in the evaluation process, i.e., at the individual level and at the fitness case level. This parallelization scheme guarantees adequate scalability as the number of cores in the GPU increases.

To compare GMGP to other GP methodologies for GPUs found in the literature, we implemented three different LGP-based and quantum-inspired approaches: (i) *compilation* (Compiler), which generates the individuals in GPU code and requires compilation; (ii) *pseudo-assembly* (Pseudo-Assembly), which generates the individuals in an intermediary assembly code and also requires compilation; and (iii) *interpretation* of multiple programs (Interpreter), which interprets the codes and does not require compilation. Our results demonstrated that GMGP outperformed all of the previous methodologies for the larger data sets of the *Mexican Hat* and *Salutowicz* benchmarks. The maximum speedups obtained were 827.7 against Compiler, 199 against Pseudo-Assembly and 7.3 against Interpreter. In terms of the GPops, for the entire evolution (GPU and CPU), GMGP achieved approximately 200.5 billion GPops for the *Mexican Hat* and *Salutowicz* benchmarks, 3.59 billion GPops for the *Mackey-Glass* benchmark, 249.9 billion GPops for the *Sobel filter* benchmark, and 2.74 trillion GPops for the *20-bit Boolean Multiplexer* benchmark.

These results provide a new perspective on GPU-based implementations of GP. Our methodology is scalable and introduces the possibility of addressing large problems within a reasonable period of time. We were the first to evolve the *20-bit Boolean Multiplexer* problem using all of the fitness cases during the evolution. The largest evolved Multiplexer that used all fitness cases in the evolution used only 11 bits, whereas the others used samples to evolve larger problems.

In our future work, we intend to develop a GP evolutionary model to run entirely in the GPU, which would offer two advantages. First, the GP model would run faster after being parallelized to GPUs. Second, we would eliminate the overhead associated with copying

the fitness results from the GPU to the CPU through the PCIe bus, yielding considerable speedups.

Acknowledgments

We would like to acknowledge support for this project from the National Counsel of Technological and Scientific Development (CNPq) and the Carlos Chagas Filho Research Support Foundation (FAPERJ).

References

Fawzan S. Alfares and Ibrahim I. Esat. Real-coded quantum inspired evolution algorithm applied to engineering optimization problems. In *Second International Symposium on Leveraging Applications of Formal Methods, Verification and Validation, ISoLA 2006*, pages 169–176. IEEE, 2006.

David Andre and John R. Koza. *Parallel Genetic Programming: a Scalable Implementation Using the Transputer Network Architecture*, incollection 16, pages 317–338. MIT Press, Cambridge, MA, USA, 1996.

Wolfgang Banzhaf, Peter Nordin, Robert E Keller, and Frank D Francone. *Genetic Programming: an Introduction: on the Automatic Evolution of Computer Programs and its Applications*. The Morgan Kaufmann Series in Artificial Intelligence. Morgan Kaufmann Publishers, 1997.

Forrest H Bennett III, John R. Koza, James Shipman, and Oscar Stiffelman. Building a parallel computer system for $18,000 that performs a half peta-flop per day. In *Proceedings of the Genetic and Evolutionary Computation Conference*, volume 2, pages 1484–1490, Orlando, Florida, USA, 1999. Morgan Kaufmann.

Markus Brameier and Wolfgang Banzhaf. *Linear Genetic Programming*. Genetic and Evolutionary Computation. Springer-Verlag, 2007.

Jens Busch, Jens Ziegler, Christian Aue, Andree Ross, Daniel Sawitzki, and Wolfgang Banzhaf. Automatic generation of control programs for walking robots using genetic programming. In *Genetic Programming*, pages 258–267. Springer, 2002.

Darren M Chitty. A data parallel approach to genetic programming using programmable graphics hardware. In *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*, pages 1566–1573. ACM, 2007.

Leandro F. Cupertino, Cleomar P. Silva, Douglas M. Dias, Marco A. C. Pacheco, and Cristiana Bentes. Evolving CUDA PTX programs by quantum inspired linear genetic programming. In *Proceedings of the 13th Annual Conference Companion on Genetic and Evolutionary Computation*, pages 399–406. ACM, 2011.

Douglas M. Dias and Marco A. C. Pacheco. Toward a quantum-inspired linear genetic programming model. In *Congress on Evolutionary Computation*, pages 1691–1698, 2009.

Douglas M. Dias and Marco A. C. Pacheco. Quantum-inspired linear genetic programming as a knowledge management system. *The Computer Journal*, 56(9):1043–1062, 2013.

Amer Draa, Souham Meshoul, Hichem Talbi, and Mohamed Batouche. A quantum inspired differential evolution algorithm for rigid image registration. In *Proceedings of the International Conference on Computational Intelligence, Istanbul*, pages 408–411, 2004.

GNU GIMP. Image manipulation program. *User Manual, Edge-Detect Filters, Sobel, The GIMP Documentation Team*, 2008.

Kuk-Hyun Han and Jong-Hwan Kim. Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. *IEEE Transactions on Evolutionary Computation*, 6(6): 580–593, 2002.

Simon Harding and Wolfgang Banzhaf. Fast genetic programming on GPUs. In *Genetic Programming*, volume 4445 of *Lecture Notes in Computer Science*, pages 90–101. Springer, 2007.

Simon Harding and Wolfgang Banzhaf. Genetic programming on GPUs for image processing. *International Journal of High Performance Systems Architecture*, 1(4):231–240, 2008.

Simon L. Harding and Wolfgang Banzhaf. Distributed genetic programming on GPUs using CUDA. In *Workshop on Parallel Architectures and Bioinspired Algorithms*, Raleigh, USA, 2009.

Jyh-Shing Roger Jang and Chuen-Tsai Sun. Predicting chaotic time series with fuzzy if-then rules. In *Second IEEE International Conference on Fuzzy Systems*, pages 1079–1084. IEEE, 1993.

John R. Koza. *Genetic Programming: on the Programming of Computers by Means of Natural Selection*. The MIT press, 1992.

John R. Koza. *Genetic Programming II: Automatic Discovery of Reusable Programs*. The MIT press, 1994.

Junjie Lai and André Seznec. Performance upper bound analysis and optimization of SGEMM on Fermi and Kepler GPUs. In *International Symposium on Code Generation and Optimization, CGO'13*, pages 1–10. IEEE Computer Society, 2013.

William B. Langdon. Large scale bioinformatics data mining with parallel genetic programming on graphics processing units. In Francisco Fernández Vega and Erick Cantú-Paz, editors, *Parallel and Distributed Computational Intelligence*, volume 269 of *Studies in Computational Intelligence*, pages 113–141. Springer Berlin Heidelberg, 2010a.

William B. Langdon. A many threaded CUDA interpreter for genetic programming. In *Proceedings of the 13th European Conference on Genetic Programming*, EuroGP'10, pages 146–158, Berlin, Heidelberg, 2010b. Springer-Verlag.

William B. Langdon. Minimising testing in genetic programming. Technical Report RN/11/10, University College London, April 2011.

William B. Langdon and Wolfgang Banzhaf. A SIMD interpreter for genetic programming on GPU graphics cards. In *Genetic Programming*, volume 4971 of *Lecture Notes in Computer Science*, pages 73–85. Springer, 2008a.

William B. Langdon and Wolfgang Banzhaf. Repeated patterns in genetic programming. *Natural Computing*, 7(4):589–613, 2008b.

William B. Langdon and Wolfgang Banzhaf. A SIMD interpreter for genetic programming on GPU graphics cards. In *Proceedings of the 11th European Conference on Genetic Programming*, EuroGP'08, pages 73–85, Berlin, Heidelberg, 2008c. Springer-Verlag.

William B. Langdon and M. Harman. Evolving a CUDA kernel from an nVidia template. In *IEEE Congress on Evolutionary Computation*, pages 1 –8, 2010.

William B. Langdon and A. Harrison. GP on SPMD parallel graphics hardware for mega bioinformatics data mining. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 12:1169–1183, 2008.

Benjamin P Lanyon, Marco Barbieri, Marcelo P. Almeida, Thomas Jennewein, Timothy C. Ralph, Kevin J. Resch, Geoff J. Pryde, Jeremy L. O'Brien, Alexei Gilchrist, and Andrew G. White. Quantum computing using shortcuts through higher dimensions. *Nature Physics 5, 134 (2009)*, 2008.

Tony E. Lewis and George D. Magoulas. Identifying similarities in TMBL programs with alignment to quicken their compilation for GPUs: computational intelligence on consumer games and graphics hardware. In *Proceedings of the 13th Annual Conference Companion on Genetic and Evolutionary Computation*, pages 447–454. ACM, 2011.

Mark Moore and Ajit Narayanan. Quantum-inspired computing. Research report 341, Department of Computer Science, University of Exeter, 1995.

Peter Nordin. AIMGP: A formal description. In *Late Breaking Papers at the Genetic Programming 1998 Conference*, University of Wisconsin, Madison, WI, USA, 1998. Stanford University Bookstore.

Nvidia. *CUDA C Programming Guide: Design Guide*. Nvidia Corporation, July 2013. Manual ID: PG-02829-001_v5.5.

Mihai Oltean, Crina Groşan, Laura Dioşan, and Cristina Mihăilă. Genetic programming with linear representation: a survey. *International Journal on Artificial Intelligence Tools*, 18(02):197–238, 2009.

Jonathan Page, Riccardo Poli, and William B. Langdon. Smooth uniform crossover with smooth point mutation in genetic programming: a preliminary study. In *Second European Workshop on Genetic Programming*, pages 39–48, London, UK, 1999. Springer-Verlag.

Michaël Defoin Platel, Stefan Schliebs, and Nikola Kasabov. Quantum-inspired evolutionary algorithm: a multimodel EDA. *IEEE Transactions on Evolutionary Computation*, 13(6): 1218–1232, 2009.

Riccardo Poli, William B. Langdon, and Nicholas Freitag McPhee. *A Field Guide to Genetic Programming*. Published via `http://lulu.com`, 2008. (contributions by J. R. Koza).

Petr Pospichal, Eoin Murphy, Michael O'Neill, Josef Schwarz, and Jiri Jaros. Acceleration of grammatical evolution using graphics processing units: computational intelligence on consumer games and graphics hardware. In *Proceedings of the 13th Annual Conference Companion on Genetic and Evolutionary Computation*, pages 431–438. ACM, 2011.

Denis Robilliard, Virginie Marion, and Cyril Fonlupt. High performance genetic programming on GPU. In *Proceedings of the 2009 Workshop on Bio-Inspired Algorithms for Distributed Systems*, BADS '09, pages 85–94, New York, NY, USA, 2009. ACM.

Abdel Salhi, Hugh Glaser, and David De Roure. Parallel implementation of a genetic-programming based tool for symbolic regression. *Information Processing Letters*, 66(6): 299–307, 1998.

Luciano R. Silveira, Ricardo Tanscheit, and Marley Vellasco. Quantum-inspired genetic algorithms applied to ordering combinatorial optimization problems. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 1–7, 2012.

Walter A. Tackett. Genetic programming for feature discovery and image discrimination. In *Proceedings of the 5th International Conference on Genetic Algorithms, ICGA-93*, pages 303–309, 1993.

Hichem Talbi, Mohamed Batouche, and Amer Draa. A quantum-inspired evolutionary algorithm for multiobjective image segmentation. *International Journal of Mathematical, Physical and Engineering Science*, 1(2):109–114, 2007.

Ian Turton, Stan Openshaw, and Gary Diplock. Some geographic applications of genetic programming on the Cray T3D supercomputer. In *UK Parallel'96*, pages 135–150, University of Surrey, 1996. Springer.

Ekaterina J Vladislavleva, Guido F Smits, and Dick Den Hertog. Order of nonlinearity as a complexity measure for models generated by symbolic regression via pareto genetic programming. *IEEE Transactions on Evolutionary Computation*, 13(2):333–349, 2009.

Allan G. Weber. The USC-SIPI image database. Technical report, University of Southern California, Signal and Image Processing Institute, Department of Electrical Engineering, Los Angeles, CA 90089-2564 USA, 3740 McClintock Ave, October 1997.

Garnett Wilson and Wolfgang Banzhaf. Linear genetic programming GPGPU on Microsoft's Xbox 360. In *Congress on Evolutionary Computation*, pages 378–385, 2008.

Gexiang Zhang. Quantum-inspired evolutionary algorithms: a survey and empirical study. *Journal of Heuristics*, 17(3):303–351, June 2011.

Journal of Machine Learning Research 16 (2015) 713-745 Submitted 4/14; Published 4/15

A Compression Technique for Analyzing Disagreement-Based Active Learning

Yair Wiener YAIR.WIENER@GMAIL.COM
Computer Science Department
Technion – Israel Institute of Technology
Haifa 32000, Israel
Steve Hanneke STEVE.HANNEKE@GMAIL.COM
Princeton, NJ 08542 USA
Ran El-Yaniv RANI@CS.TECHNION.AC.IL
Computer Science Department
Technion – Israel Institute of Technology
Haifa 32000, Israel

Editor: Sanjoy Dasgupta

Abstract

We introduce a new and improved characterization of the label complexity of disagreement-based active learning, in which the leading quantity is the *version space compression set size*. This quantity is defined as the size of the smallest subset of the training data that induces the same version space. We show various applications of the new characterization, including a tight analysis of CAL and refined label complexity bounds for linear separators under mixtures of Gaussians and axis-aligned rectangles under product densities. The version space compression set size, as well as the new characterization of the label complexity, can be naturally extended to agnostic learning problems, for which we show new speedup results for two well known active learning algorithms.

Keywords: active learning, selective sampling, sequential design, statistical learning theory, PAC learning, sample complexity, selective prediction

1. Introduction

Active learning is a learning paradigm allowing the learner to sequentially request the target labels of selected instances from a pool or stream of unlabeled data.[1] The key question in the theoretical analysis of active learning is how many label requests are sufficient to learn the labeling function to a specified accuracy, a quantity known as the *label complexity*. Among the many recent advances in the theory of active learning, perhaps the most well-studied technique has been the *disagreement-based* approach, initiated by Cohn, Atlas, and Ladner (1994), and further advanced in numerous articles (e.g., Balcan, Beygelzimer, and Langford, 2009; Dasgupta, Hsu, and Monteleoni, 2007; Beygelzimer, Dasgupta, and Langford, 2009; Beygelzimer, Hsu, Langford, and Zhang, 2010; Koltchinskii, 2010; Hanneke, 2012; Hanneke and Yang, 2012). The basic strategy in disagreement-based active learning is to sequentially process the unlabeled examples, and for each example, the algorithm requests its label if and only if the value of the optimal classifier's classification on that point cannot be inferred from information already obtained.

1. Any active learning technique for streaming data can be used in pool-based models but not vice versa

One attractive feature of this approach is that its simplicity makes it amenable to thorough theoretical analysis, and numerous theoretical guarantees on the performance of variants of this strategy under various conditions have appeared in the literature (see e.g., Balcan, Beygelzimer, and Langford, 2009; Hanneke, 2007a; Dasgupta, Hsu, and Monteleoni, 2007; Balcan, Broder, and Zhang, 2007; Beygelzimer, Dasgupta, and Langford, 2009; Friedman, 2009; Balcan, Hanneke, and Vaughan, 2010; Hanneke, 2011; Koltchinskii, 2010; Beygelzimer, Hsu, Langford, and Zhang, 2010; Hsu, 2010; Hanneke, 2012; El-Yaniv and Wiener, 2012; Hanneke and Yang, 2012; Hanneke, 2014). The majority of these results formulate bounds on the label complexity in terms of a complexity measure known as the *disagreement coefficient* (Hanneke, 2007a), which we define below. A notable exception to this is the recent work of El-Yaniv and Wiener (2012), rooted in the related topic of selective prediction (El-Yaniv and Wiener, 2010; Wiener and El-Yaniv, 2012; Wiener, 2013; Wiener and El-Yaniv, 2015), which instead bounds the label complexity in terms of two complexity measures called the *characterizing set complexity* and the *version space compression set size* (El-Yaniv and Wiener, 2010). In the current literature, the above are the only known general techniques for the analysis of disagreement-based active learning.

In the present article, we present a new characterization of the label complexity of disagreement-based active learning. The leading quantity in our characterization is the *version space compression set size* of El-Yaniv and Wiener (2012, 2010); Wiener (2013), which corresponds to the size of the smallest subset of the training set that induces the same version space as the entire training set. This complexity measure was shown by El-Yaniv and Wiener (2012) to be a special case of the extended teaching dimension of Hanneke (2007b).

The new characterization improves upon the two prior techniques in some cases. For a noiseless setting (the realizable case), we show that the label complexity results derived from this new technique are *tight* up to logarithmic factors. This was not true of either of the previous techniques; as we discuss in Appendix B, the known upper bounds in the literature expressed in terms of these other complexity measures are sometimes off by a factor of the VC dimension. Moreover, the new method significantly simplifies the recent technique of Wiener (2013); El-Yaniv and Wiener (2012, 2010) by completely eliminating the need for the characterizing set complexity measure.

Interestingly, interpreted as an upper bound on the label complexity of active learning in general, the upper bounds presented here also reflect improvements over a bound of Hanneke (2007b), which is also expressed in terms of (a target-independent variant of) this same complexity measure: specifically, reducing the bound by roughly a factor of the VC dimension compared to that result. In addition to these results on the label complexity, we also relate the version space compression set size to the disagreement coefficient, essentially showing that they are always within a factor of the VC dimension of each other (with additional logarithmic factors).

We apply this new technique to derive new results for two learning problems: namely, linear separators under mixtures of Gaussians, and axis-aligned hyperrectangles under product densities. We derive bounds on the version space compression set size for each of these. Thus, using our results relating the version space compression set size to the label complexity, we arrive at bounds on the label complexity of disagreement-based active learning for these problems, which represent significant refinements of the best results in the prior literature on these settings.

While the version space compression set size is initially defined for noiseless (realizable) learning problems that have a version space, it can be naturally extended to an agnostic setting, and the new technique applies to noisy, agnostic problems as well. This surprising result, which was motivated by related observations of Hanneke (2014); Wiener (2013), is allowed through bounds on

the disagreement coefficient in terms of the version space compression set size, and the applicability of the disagreement coefficient to both the realizable and agnostic settings. We formulate this generalization in Section 6 and present new sample complexity results for known active learning algorithms, including the disagreement-based methods of Dasgupta, Hsu, and Monteleoni (2007) and Hanneke (2012). These results tighten the bounds of Wiener (2013) using the new technique.

2. Preliminary Definitions

Let $\mathcal{X}$ denote a set, called the *instance space*, and let $\mathcal{Y} \triangleq \{-1,+1\}$, called the *label space*. A *classifier* is a measurable function $h : \mathcal{X} \to \mathcal{Y}$. Throughout, we fix a set $\mathcal{F}$ of classifiers, called the *concept space*, and denote by d the VC dimension of $\mathcal{F}$ (Vapnik and Chervonenkis, 1971; Vapnik, 1998). We also fix an arbitrary probability measure P over $\mathcal{X} \times \mathcal{Y}$, called the *data distribution*. Aside from Section 6, we make the assumption that $\exists f^* \in \mathcal{F}$ with $\mathbb{P}(Y = f^*(x)|X = x) = 1$ for all $x \in \mathcal{X}$, where $(X,Y) \sim P$; this is known as the *realizable case*, and f^* is known as the *target function*. For any classifier h, define its *error rate* $\mathrm{er}(h) \triangleq P((x,y) : h(x) \neq y)$; note that $\mathrm{er}(f^*) = 0$.

For any set $\mathcal{H}$ of classifiers, define the *region of disagreement*

$$\mathrm{DIS}(\mathcal{H}) \triangleq \{x \in \mathcal{X} : \exists h, g \in \mathcal{H} \text{ s.t. } h(x) \neq g(x)\}.$$

Also define $\Delta\mathcal{H} \triangleq P(\mathrm{DIS}(\mathcal{H}) \times \mathcal{Y})$, the marginal probability of the region of disagreement.

Let $S_\infty \triangleq \{(x_1,y_1),(x_2,y_2),\ldots\}$ be a sequence of i.i.d. P-distributed random variables, and for each $m \in \mathbb{N}$, denote by $S_m \triangleq \{(x_1,y_1),\ldots,(x_m,y_m)\}$.[2] For any $m \in \mathbb{N} \cup \{0\}$, and any $S \in (\mathcal{X} \times \mathcal{Y})^m$, define the *version space* $\mathrm{VS}_{\mathcal{F},S} \triangleq \{h \in \mathcal{F} : \forall (x,y) \in S, h(x) = y\}$ (Mitchell, 1977). The following definition will be central in our results below.

Definition 1 (Version Space Compression Set Size) *For any $m \in \mathbb{N} \cup \{0\}$ and any $S \in (\mathcal{X} \times \mathcal{Y})^m$, the* version space compression set *$\hat{\mathcal{C}}_S$ is a smallest subset of S satisfying $\mathrm{VS}_{\mathcal{F},\hat{\mathcal{C}}_S} = \mathrm{VS}_{\mathcal{F},S}$. The* version space compression set size *is defined to be $\hat{n}(\mathcal{F},S) \triangleq |\hat{\mathcal{C}}_S|$. In the special cases where $\mathcal{F}$ and perhaps $S = S_m$ are obvious from the context, we abbreviate $\hat{n} \triangleq \hat{n}(S_m) \triangleq \hat{n}(\mathcal{F},S_m)$.*

Note that the value $\hat{n}(\mathcal{F},S)$ is unique for any S, and $\hat{n}(S_m)$ is, obviously, a random number that depends on the (random) sample S_m. The quantity $\hat{n}(S_m)$ has been studied under at least two names in the prior literature. Drawing motivation from the work on Exact learning with Membership Queries (Hegedüs, 1995; Hellerstein, Pillaipakkamnatt, Raghavan, and Wilkins, 1996), which extends ideas from Goldman and Kearns (1995) on the complexity of teaching, the quantity $\hat{n}(S_m)$ was introduced in the work of Hanneke (2007b) as the *extended teaching dimension* of the classifier f^* on the space $\{x_1,\ldots,x_m\}$ with respect to the set $\mathcal{F}[\{x_1,\ldots,x_m\}] \triangleq \{x_i \mapsto h(x_i) : h \in \mathcal{F}\}$ of distinct classifications of $\{x_1,\ldots,x_m\}$ realized by $\mathcal{F}$; in this context, the set $\hat{\mathcal{C}}_{S_m}$ is known as a *minimal specifying set* of f^* on $\{x_1,\ldots,x_m\}$ with respect to $\mathcal{F}[\{x_1,\ldots,x_m\}]$. The quantity $\hat{n}(S_m)$ was independently discovered by El-Yaniv and Wiener (2010) in the context of selective classification, which is the source of the compression set terminology introduced above; we adopt this terminology throughout the present article. See the work of El-Yaniv and Wiener (2012) for a formal proof of the equivalence of these two notions.

It will also be useful to define minimal confidence bounds on certain quantities, as follows.

2. Note that, in the realizable case, $y_i = f^*(x_i)$ for all i with probability 1. For simplicity, we will suppose these equalities hold throughout our discussion of the realizable case.

Definition 2 (Version Space Compression Set Size Minimal Bound) *For any $m \in \mathbb{N} \cup \{0\}$ and $\delta \in (0,1]$, define the* version space compression set size minimal bound

$$\mathcal{B}_{\hat{n}}(m,\delta) \triangleq \min\left\{b \in \mathbb{N} \cup \{0\} : \mathbb{P}(\hat{n}(S_m) \leq b) \geq 1-\delta\right\}.$$

Similarly, define the *version space disagreement region minimal bound*

$$\mathcal{B}_{\Delta}(m,\delta) \triangleq \min\left\{t \in [0,1] : \mathbb{P}(\Delta \mathrm{VS}_{\mathcal{F},S_m} \leq t) \geq 1-\delta\right\}.$$

In both cases, the quantities implicitly also depend on $\mathcal{F}$ and P (which remain fixed throughout our analysis below), and the only random variables involved in these probabilities are the data S_m.

Most of the existing general results on disagreement-based active learning are expressed in terms of a quantity known as the *disagreement coefficient* (Hanneke, 2007a, 2009), defined as follows.

Definition 3 (Disagreement Coefficient) *For any classifier f and $r > 0$, define the r-ball centered at f as*

$$\mathrm{B}(f,r) \triangleq \{h \in \mathcal{F} : \Delta\{h,f\} \leq r\},$$

and for any $r_0 \geq 0$, define the disagreement coefficient *of $\mathcal{F}$ with respect to P as*[3]

$$\theta(r_0) \triangleq \sup_{r > r_0} \frac{\Delta \mathrm{B}(f^*,r)}{r} \vee 1.$$

The disagreement coefficient was originally introduced to the active learning literature by Hanneke (2007a), and has been studied and bounded by a number of authors (see e.g., Hanneke, 2007a; Friedman, 2009; Wang, 2011; Hanneke, 2014; Balcan and Long, 2013). Similar quantities have also been studied in the passive learning literature, rooted in the work of Alexander (see e.g., Alexander, 1987; Giné and Koltchinskii, 2006).

Numerous recent results, many of which are surveyed by Hanneke (2014), exhibit bounds on the label complexity of disagreement-based active learning in terms of the disagreement coefficient. It is therefore of major interest to develop such bounds for specific cases of interest (i.e., for specific classes $\mathcal{F}$ and distributions P). In particular, any result showing $\theta(r_0) = o(1/r_0)$ indicates that disagreement-based active learning should asymptotically provide some advantage over passive learning for that $\mathcal{F}$ and P (Hanneke, 2012). We are particularly interested in scenarios in which $\theta(r_0) = O(\mathrm{polylog}(1/r_0))$, or even $\theta(r_0) = O(1)$, since these imply strong improvements over passive learning (Hanneke, 2007a, 2011).

There are several general results on the asymptotic behavior of the disagreement coefficient as $r_0 \to 0$, for interesting cases. For the class of linear separators in $\mathbb{R}^k$, perhaps the most general result to date is that the existence of a density function for the marginal distribution of P over $\mathcal{X}$ is sufficient to guarantee $\theta(r_0) = o(1/r_0)$ (Hanneke, 2014). That work also shows that, if the density is bounded and has bounded support, and the target separator passes through the support at a continuity point of the density, then $\theta(r_0) = O(1)$. In both of these cases, for $k \geq 2$, the specific dependence on r_0 in the little-o and the constant factors in the big-O will vary depending on the particular distribution P, and in particular, will depend on f^* (i.e., such bounds are *target-dependent*).

There are also several explicit, *target-independent* bounds on the disagreement coefficient in the literature. Perhaps the most well-known of these is for homogeneous linear separators in $\mathbb{R}^k$, where

3. We use the notation $a \vee b = \max\{a,b\}$.

the marginal distribution of P over $\mathcal{X}$ is confined to be the uniform distribution over the unit sphere, in which case $\theta(r_0)$ is known to be within a factor of 4 of $\min\{\pi\sqrt{k}, 1/r_0\}$ (Hanneke, 2007a). In the present paper, we are primarily focused on explicit, target-independent speedup bounds, though our abstract results can be used to derive bounds of either type.

3. Relating $\hat{n}$ and the Disagreement Coefficient

In this section, we show how to bound the disagreement coefficient in terms of $\mathcal{B}_{\hat{n}}(m,\delta)$. We also show the other direction and bound $\mathcal{B}_{\hat{n}}(m,\delta)$ in terms of the disagreement coefficient.

Theorem 4 *For any $r_0 \in (0,1)$,*

$$\theta(r_0) \leq \max\left\{\max_{r\in(r_0,1)} 16\mathcal{B}_{\hat{n}}\left(\left\lceil\frac{1}{r}\right\rceil, \frac{1}{20}\right), 512\right\}.$$

Proof We will prove that, for any $r \in (0,1)$,

$$\frac{\Delta \mathrm{B}(f^*, r)}{r} \leq \max\left\{16\mathcal{B}_{\hat{n}}\left(\left\lceil\frac{1}{r}\right\rceil, \frac{1}{20}\right), 512\right\}. \tag{1}$$

The result then follows by taking the supremum of both sides over $r \in (r_0, 1)$.

Fix $r \in (0,1)$, let $m = \lceil 1/r \rceil$, and for $i \in \{1,\ldots,m\}$, define $S_{m\setminus i} = S_m \setminus \{(x_i, y_i)\}$. Also define $D_{m\setminus i} = \mathrm{DIS}(\mathrm{VS}_{\mathcal{F},S_{m\setminus i}} \cap \mathrm{B}(f^*, r))$ and $\Delta_{m\setminus i} = \mathbb{P}(x_i \in D_{m\setminus i} | S_{m\setminus i}) = P(D_{m\setminus i} \times \mathcal{Y})$. If $\Delta\mathrm{B}(f^*, r)m \leq 512$, (1) clearly holds. Otherwise, suppose $\Delta\mathrm{B}(f^*, r)m > 512$. If $x_i \in \mathrm{DIS}(\mathrm{VS}_{\mathcal{F},S_{m\setminus i}})$, then we must have $(x_i, y_i) \in \hat{C}_{S_m}$. So

$$\hat{n}(S_m) \geq \sum_{i=1}^{m} \mathbb{1}_{\mathrm{DIS}(\mathrm{VS}_{\mathcal{F},S_{m\setminus i}})}(x_i).$$

Therefore,

$$\begin{aligned}
&\mathbb{P}\{\hat{n}(S_m) \leq (1/16)\Delta\mathrm{B}(f^*, r)m\} \\
&\leq \mathbb{P}\left\{\sum_{i=1}^{m} \mathbb{1}_{\mathrm{DIS}(\mathrm{VS}_{\mathcal{F},S_{m\setminus i}})}(x_i) \leq (1/16)\Delta\mathrm{B}(f^*, r)m\right\} \\
&\leq \mathbb{P}\left\{\sum_{i=1}^{m} \mathbb{1}_{D_{m\setminus i}}(x_i) \leq (1/16)\Delta\mathrm{B}(f^*, r)m\right\} \\
&= \mathbb{P}\left\{\sum_{i=1}^{m} \mathbb{1}_{\mathrm{DIS}(\mathrm{B}(f^*,r))}(x_i) - \mathbb{1}_{D_{m\setminus i}}(x_i) \geq \sum_{i=1}^{m} \mathbb{1}_{\mathrm{DIS}(\mathrm{B}(f^*,r))}(x_i) - (1/16)\Delta\mathrm{B}(f^*, r)m\right\}.
\end{aligned}$$

Breaking the above event into two cases based on the value of $\sum_{i=1}^m \mathbb{1}_{\mathrm{DIS}(\mathrm{B}(f^*,r))}(x_i)$, this last line equals

$$\begin{aligned}
&\mathbb{P}\left\{\sum_{i=1}^m \mathbb{1}_{\mathrm{DIS}(\mathrm{B}(f^*,r))}(x_i) - \mathbb{1}_{D_{m\setminus i}}(x_i) \geq \right. \\
&\qquad \left. \sum_{i=1}^m \mathbb{1}_{\mathrm{DIS}(\mathrm{B}(f^*,r))}(x_i) - \frac{1}{16}\Delta\mathrm{B}(f^*,r)m, \quad \sum_{i=1}^m \mathbb{1}_{\mathrm{DIS}(\mathrm{B}(f^*,r))}(x_i) < \frac{7}{8}\Delta\mathrm{B}(f^*,r)m \right\} \\
&+\mathbb{P}\left\{\sum_{i=1}^m \mathbb{1}_{\mathrm{DIS}(\mathrm{B}(f^*,r))}(x_i) - \mathbb{1}_{D_{m\setminus i}}(x_i) \geq \right. \\
&\qquad \left. \sum_{i=1}^m \mathbb{1}_{\mathrm{DIS}(\mathrm{B}(f^*,r))}(x_i) - \frac{1}{16}\Delta\mathrm{B}(f^*,r)m, \quad \sum_{i=1}^m \mathbb{1}_{\mathrm{DIS}(\mathrm{B}(f^*,r))}(x_i) \geq \frac{7}{8}\Delta\mathrm{B}(f^*,r)m \right\} \\
&\leq \mathbb{P}\left\{\sum_{i=1}^m \mathbb{1}_{\mathrm{DIS}(\mathrm{B}(f^*,r))}(x_i) < (7/8)\Delta\mathrm{B}(f^*,r)m\right\} \\
&\quad +\mathbb{P}\left\{\sum_{i=1}^m \mathbb{1}_{\mathrm{DIS}(\mathrm{B}(f^*,r))}(x_i) - \mathbb{1}_{D_{m\setminus i}}(x_i) \geq (13/16)\Delta\mathrm{B}(f^*,r)m\right\}.
\end{aligned}$$

Since we are considering the case $\Delta\mathrm{B}(f^*,r)m > 512$, a Chernoff bound implies

$$\mathbb{P}\left(\sum_{i=1}^m \mathbb{1}_{\mathrm{DIS}(\mathrm{B}(f^*,r))}(x_i) < (7/8)\Delta\mathrm{B}(f^*,r)m\right) \leq \exp\{-\Delta\mathrm{B}(f^*,r)m/128\} < e^{-4}.$$

Furthermore, Markov's inequality implies

$$\mathbb{P}\left(\sum_{i=1}^m \mathbb{1}_{\mathrm{DIS}(\mathrm{B}(f^*,r))}(x_i) - \mathbb{1}_{D_{m\setminus i}}(x_i) \geq (13/16)\Delta\mathrm{B}(f^*,r)m\right) \leq \frac{m\Delta\mathrm{B}(f^*,r) - \mathbb{E}\left[\sum_{i=1}^m \mathbb{1}_{D_{m\setminus i}}(x_i)\right]}{(13/16)m\Delta\mathrm{B}(f^*,r)}.$$

Since the x_i values are exchangeable,

$$\mathbb{E}\left[\sum_{i=1}^m \mathbb{1}_{D_{m\setminus i}}(x_i)\right] = \sum_{i=1}^m \mathbb{E}\left[\mathbb{E}\left[\mathbb{1}_{D_{m\setminus i}}(x_i)\middle| S_{m\setminus i}\right]\right] = \sum_{i=1}^m \mathbb{E}\left[\Delta_{m\setminus i}\right] = m\mathbb{E}\left[\Delta_{m\setminus m}\right].$$

Hanneke (2012) proves that this is at least

$$m(1-r)^{m-1}\Delta\mathrm{B}(f^*,r).$$

In particular, when $\Delta\mathrm{B}(f^*,r)m > 512$, we must have $r < 1/511 < 1/2$, which implies $(1-r)^{\lceil 1/r\rceil - 1} \geq 1/4$, so that we have

$$\mathbb{E}\left[\sum_{i=1}^m \mathbb{1}_{D_{m\setminus i}}(x_i)\right] \geq (1/4)m\Delta\mathrm{B}(f^*,r).$$

Altogether, we have established that

$$\mathbb{P}\left(\hat{n}(S_m) \leq (1/16)\Delta\mathrm{B}(f^*,r)m\right) < \frac{m\Delta\mathrm{B}(f^*,r) - (1/4)m\Delta\mathrm{B}(f^*,r)}{(13/16)m\Delta\mathrm{B}(f^*,r)} + e^{-4} = \frac{12}{13} + e^{-4} < \frac{19}{20}.$$

Thus, since $\hat{n}(S_m) \leq \mathcal{B}_{\hat{n}}\left(m, \frac{1}{20}\right)$ with probability at least $\frac{19}{20}$, we must have that

$$\mathcal{B}_{\hat{n}}\left(m, \frac{1}{20}\right) > (1/16)\Delta \mathrm{B}(f^*, r)m \geq (1/16)\frac{\Delta \mathrm{B}(f^*, r)}{r}.$$

■

The following Theorem, whose proof is given in Section 4, is a "converse" of Theorem 4, showing a bound on $\mathcal{B}_{\hat{n}}(m, \delta)$ in terms of the disagreement coefficient.

Theorem 5 *There is a finite universal constant $c > 0$ such that, $\forall r_0, \delta \in (0,1)$,*

$$\max_{r \in (r_0, 1)} \mathcal{B}_{\hat{n}}\left(\left\lceil \frac{1}{r} \right\rceil, \delta\right) \leq c\theta(dr_0)\left(d\ln(e\theta(dr_0)) + \ln\left(\frac{\log_2(2/r_0)}{\delta}\right)\right)\log_2\left(\frac{2}{r_0}\right).$$

4. A Tight Analysis of CAL

The following algorithm is due to Cohn, Atlas, and Ladner (1994).

Algorithm: CAL(n)
0. $m \leftarrow 0, t \leftarrow 0, V_0 \leftarrow \mathcal{F}$
1. While $t < n$
2. $\quad m \leftarrow m + 1$
3. $\quad$ If $x_m \in \mathrm{DIS}(V_{m-1})$
4. $\quad\quad$ Request label y_m; let $V_m \leftarrow \{h \in V_{m-1} : h(x_m) = y_m\}$, $t \leftarrow t + 1$
5. $\quad$ Else $V_m \leftarrow V_{m-1}$
6. Return any $\hat{h} \in V_m$

One particularly attractive feature of this algorithm is that it maintains the invariant that $V_m = \mathrm{VS}_{\mathcal{F}, S_m}$ for all values of m it obtains (since, if $V_{m-1} = \mathrm{VS}_{\mathcal{F}, S_{m-1}}$, then $f^* \in V_{m-1}$, so any point $x_m \notin \mathrm{DIS}(V_{m-1})$ has $\{h \in V_{m-1} : h(x_m) = y_m\} = \{h \in V_{m-1} : h(x_m) = f^*(x_m)\} = V_{m-1}$ anyway). To analyze this method, we first define, for every $m \in \mathbb{N}$,

$$N(m; S_m) = \sum_{t=1}^{m} \mathbb{1}_{\mathrm{DIS}(\mathrm{VS}_{\mathcal{F}, S_{t-1}})}(x_t),$$

which counts the number of labels requested by CAL among the first m data points (assuming it does not halt first). The following result provides data-dependent upper and lower bounds on this important quantity, which will be useful in establishing label complexity bounds for CAL below.

Lemma 6

$$\max_{t \leq m} \hat{n}(S_t) \leq N(m; S_m),$$

and with probability at least $1 - \delta$,

$$N(m; S_m) \leq \max_{t \in \{2^i : i \in \{0, \ldots, \lfloor \log_2(m) \rfloor\}\}} \left(55\hat{n}(S_t)\ln\left(\frac{et}{\hat{n}(S_t)}\right) + 24\ln\left(\frac{4\log_2(2m)}{\delta}\right)\right)\log_2(2m).$$

Since the upper and lower bounds on $N(m;S_m)$ in Lemma 6 require access to the *labels* of the data, they are not as much interesting for practice as they are for their theoretical significance. In particular, they will allow us to derive new distribution-dependent bounds on the performance of CAL below (Theorems 9 and 10). Lemma 6 is also of some *conceptual* significance, as it shows a direct and fairly-tight connection between the behavior of CAL and the size of the version space compression set.

The proof of the upper bound on $N(m;S_m)$ relies on the following two lemmas. The first lemma (Lemma 7) is implied by a classical compression bound of Littlestone and Warmuth (1986), and provides a high-confidence bound on the probability measure of a set, given that it has zero empirical frequency and is specified by a small number of samples. For completeness, we include a proof of this result below: a variant of the original argument of Littlestone and Warmuth (1986).[4]

Lemma 7 (Compression; Littlestone and Warmuth, 1986) *For any $\delta \in (0,1)$, any collection $\mathbb{D}$ of measurable sets $D \subseteq \mathcal{X} \times \mathcal{Y}$, any $m \in \mathbb{N}$ and $n \in \mathbb{N} \cup \{0\}$ with $n \leq m$, and any permutation-invariant function $\phi_n : (\mathcal{X} \times \mathcal{Y})^n \to \mathbb{D}$, with probability of at least $1-\delta$ over draw of S_m, every distinct $i_1, \ldots, i_n \in \{1, \ldots, m\}$ with $S_m \cap \phi_n((x_{i_1}, y_{i_1}), \ldots, (x_{i_n}, y_{i_n})) = \emptyset$ satisfies*[5]

$$P(\phi_n((x_{i_1}, y_{i_1}), \ldots, (x_{i_n}, y_{i_n}))) \leq \frac{1}{m-n}\left(n \ln\left(\frac{em}{n}\right) + \ln\left(\frac{1}{\delta}\right)\right). \tag{2}$$

Proof Let $\varepsilon > 0$ denote the value of the right hand side of (2). The result trivially holds if $\varepsilon > 1$. For the remainder, consider the case $\varepsilon \leq 1$. Let I_n be the set of all sets of n distinct indices $\{i_1, \ldots, i_n\}$ from $\{1, \ldots, m\}$. Note that $|I_n| = \binom{m}{n}$. Given a labeled sample S_m and $\mathbf{i} = \{i_1, \ldots, i_n\} \in I_n$, denote by $S_m^{\mathbf{i}} = \{(x_{i_1}, y_{i_1}), \ldots, (x_{i_n}, y_{i_n})\}$, and by $S_m^{-\mathbf{i}} = \{(x_i, y_i) : i \in \{1, \ldots, m\} \setminus \mathbf{i}\}$. Since ϕ_n is permutation-invariant, for any distinct $i_1, \ldots, i_n \in \{1, \ldots, m\}$, letting $\mathbf{i} = \{i_1, \ldots, i_n\}$ denote the unordered set of indices, we may denote $\phi_n(S_m^{\mathbf{i}}) = \phi_n((x_{i_1}, y_{i_1}), \ldots, (x_{i_n}, y_{i_n}))$ without ambiguity. In particular, we have $\{\phi_n((x_{i_1}, y_{i_1}), \ldots, (x_{i_n}, y_{i_n})) : i_1, \ldots, i_n \in \{1, \ldots, m\} \text{ distinct}\} = \{\phi_n(S_m^{\mathbf{i}}) : \mathbf{i} \in I_n\}$, so that it suffices to show that, with probability at least $1-\delta$, every $\mathbf{i} \in I_n$ with $S_m \cap \phi_n(S_m^{\mathbf{i}}) = \emptyset$ has $P(\phi_n(S_m^{\mathbf{i}})) \leq \varepsilon$.

Define the events $\omega(\mathbf{i}, m) = \left\{S_m \cap \phi_n(S_m^{\mathbf{i}}) = \emptyset\right\}$ and $\omega'(\mathbf{i}, m-n) = \left\{S_m^{-\mathbf{i}} \cap \phi_n(S_m^{\mathbf{i}}) = \emptyset\right\}$. Note that $\omega(\mathbf{i}, m) \subseteq \omega'(\mathbf{i}, m-n)$. Therefore, for each $\mathbf{i} \in I_n$, we have

$$\mathbb{P}\left(\left\{P(\phi_n(S_m^{\mathbf{i}})) > \varepsilon\right\} \cap \omega(\mathbf{i}, m)\right) \leq \mathbb{P}\left(\left\{P(\phi_n(S_m^{\mathbf{i}})) > \varepsilon\right\} \cap \omega'(\mathbf{i}, m-n)\right).$$

By the law of total probability and $\sigma(S_m^{\mathbf{i}})$-measurability of the event $\left\{P(\phi_n(S_m^{\mathbf{i}})) > \varepsilon\right\}$, this equals

$$\mathbb{E}\left[\mathbb{P}\left(\left\{P(\phi_n(S_m^{\mathbf{i}})) > \varepsilon\right\} \cap \omega'(\mathbf{i}, m-n) \middle| S_m^{\mathbf{i}}\right)\right] = \mathbb{E}\left[\mathbb{1}[P(\phi_n(S_m^{\mathbf{i}})) > \varepsilon]\mathbb{P}\left(\omega'(\mathbf{i}, m-n) \middle| S_m^{\mathbf{i}}\right)\right].$$

Noting that $|S_m^{-\mathbf{i}} \cap \phi_n(S_m^{\mathbf{i}})|$ is conditionally Binomial$(m-n, P(\phi_n(S_m^{\mathbf{i}})))$ given $S_m^{\mathbf{i}}$, this equals

$$\mathbb{E}\left[\mathbb{1}[P(\phi_n(S_m^{\mathbf{i}})) > \varepsilon]\left(1 - P(\phi_n(S_m^{\mathbf{i}}))\right)^{m-n}\right] \leq (1-\varepsilon)^{m-n} \leq e^{-\varepsilon(m-n)},$$

4. See also Section 5.2.1 of Herbrich (2002) for a very clear and concise proof of a similar result (beginning with the line above (5.15) there, for our purposes).
5. We define $0\ln(1/0) = 0\ln(\infty) = 0$.

where the last inequality is due to $1-\varepsilon \leq e^{-\varepsilon}$ (see e.g., Theorem A.101 of Herbrich, 2002). In the case $n=0$, this last expression equals δ, which establishes the result since $|I_0|=1$. Otherwise, if $n>0$, combining the above with a union bound, we have that

$$\mathbb{P}\left(\exists \mathbf{i} \in I_n : P(\phi_n(S_m^{\mathbf{i}})) > \varepsilon \wedge S_m \cap \phi_n(S_m^{\mathbf{i}}) = \emptyset\right) = \mathbb{P}\left(\bigcup_{\mathbf{i}\in I_n}\left\{P(\phi_n(S_m^{\mathbf{i}})) > \varepsilon\right\} \cap \omega(\mathbf{i},m)\right)$$
$$\leq \sum_{\mathbf{i}\in I_n} \mathbb{P}\left(\left\{P(\phi_n(S_m^{\mathbf{i}})) > \varepsilon\right\} \cap \omega(\mathbf{i},m)\right) \leq \sum_{\mathbf{i}\in I_n} e^{-\varepsilon(m-n)} = \binom{m}{n} e^{-\varepsilon(m-n)}.$$

Since $\binom{m}{n} \leq \left(\frac{em}{n}\right)^n$ (see e.g., Theorem A.105 of Herbrich, 2002), this last expression is at most $\left(\frac{em}{n}\right)^n e^{-\varepsilon(m-n)} = \delta$, which completes the proof. ■

The following, Lemma 8, will be used for proving Lemma 6 above. The lemma relies on Lemma 7 and provides a high-confidence bound on the probability of requesting the next label at any given point in the CAL algorithm. This refines a related result of El-Yaniv and Wiener (2010). Lemma 8 is also of independent interest in the context of selective prediction (Wiener, 2013; El-Yaniv and Wiener, 2010), as it can be used to improve the known coverage bounds for realizable selective classification.

Lemma 8 *For any $\delta \in (0,1)$ and $m \in \mathbb{N}$, with probability at least $1-\delta$,*

$$\Delta \mathrm{VS}_{\mathcal{F},S_m} \leq \frac{10\hat{n}(S_m)\ln\left(\frac{em}{\hat{n}(S_m)}\right) + 4\ln\left(\frac{2}{\delta}\right)}{m}.$$

Proof The proof is similar to that of a result of El-Yaniv and Wiener (2010), except using a generalization bound based directly on sample compression, rather than the VC dimension. Specifically, let $\mathbb{D} = \{\mathrm{DIS}(\mathrm{VS}_{\mathcal{F},S}) \times \mathcal{Y} : S \in (\mathcal{X}\times\mathcal{Y})^m\}$, and for each $n \leq m$ and $S \in (\mathcal{X}\times\mathcal{Y})^n$, let $\phi_n(S) = \mathrm{DIS}(\mathrm{VS}_{\mathcal{F},S}) \times \mathcal{Y}$. In particular, note that for any $n \geq \hat{n}(S_m)$, any superset S of $\hat{\mathcal{C}}_{S_m}$ of size n contained in S_m has $\phi_n(S) = \mathrm{DIS}(\mathrm{VS}_{\mathcal{F},S_m}) \times \mathcal{Y}$, and therefore $S_m \cap \phi_n(S) = \emptyset$ and $\Delta\mathrm{VS}_{\mathcal{F},S_m} = P(\phi_n(S))$. Therefore, Lemma 7 implies that, for each $n \in \{0,\ldots,m\}$, with probability at least $1-\delta/(n+2)^2$, if $\hat{n}(S_m) \leq n$,

$$\Delta \mathrm{VS}_{\mathcal{F},S_m} \leq \frac{1}{m-n}\left(n\ln\left(\frac{em}{n}\right) + \ln\left(\frac{(n+2)^2}{\delta}\right)\right).$$

Furthermore, since $\Delta\mathrm{VS}_{\mathcal{F},S_m} \leq 1$, any $n \geq m/2$ trivially has $\Delta\mathrm{VS}_{\mathcal{F},S_m} \leq 2n/m \leq (2/m)(n\ln(em/n) + \ln((n+2)^2/\delta))$, while any $n \leq m/2$ has $1/(m-n) \leq 2/m$, so that the above is at most

$$\frac{2}{m}\left(n\ln\left(\frac{em}{n}\right) + \ln\left(\frac{(n+2)^2}{\delta}\right)\right).$$

Additionally, $\ln((n+2)^2) \leq 2\ln(2) + 4n \leq 2\ln(2) + 4n\ln(em/n)$, so that the above is at most

$$\frac{2}{m}\left(5n\ln\left(\frac{em}{n}\right) + 2\ln\left(\frac{2}{\delta}\right)\right).$$

By a union bound, this holds for all $n \in \{0,\ldots,m\}$ with probability at least $1-\sum_{n=0}^{m}\delta/(n+2)^2 > 1-\delta$. In particular, since $\hat{n}(S_m)$ is always in $\{0,\ldots,m\}$, this implies the result. ■

Proof of Lemma 6 For any $t \leq m$, by definition of $\hat{n}$ (in particular, minimality), *any* set $S \subset S_t$ with $|S| < \hat{n}(S_t)$ necessarily has $\mathrm{VS}_{\mathcal{F},S} \neq \mathrm{VS}_{\mathcal{F},S_t}$. Thus, since CAL maintains that $V_t = \mathrm{VS}_{\mathcal{F},S_t}$, and V_t is precisely the set of classifiers in $\mathcal{F}$ that are correct on the $N(t;S_t)$ points (x_i,y_i) with $i \leq t$ for which $\mathbb{1}_{\mathrm{DIS}(\mathrm{VS}_{\mathcal{F},S_{i-1}})}(x_i) = 1$, we must have $N(t;S_t) \geq \hat{n}(S_t)$. We therefore have $\max_{t \leq m} \hat{n}(S_t) \leq \max_{t \leq m} N(t;S_t) = N(m;S_m)$ (by monotonicity of $t \mapsto N(t;S_t)$).

For the upper bound, let δ_i be a sequence of values in $(0,1]$ with $\sum_{i=0}^{\lfloor \log_2(m) \rfloor} \delta_i \leq \delta/2$. Lemma 8 implies that, for each i, with probability at least $1-\delta_i$,

$$\Delta\mathrm{VS}_{\mathcal{F},S_{2^i}} \leq 2^{-i}\left(10\hat{n}(S_{2^i})\ln\left(\frac{e2^i}{\hat{n}(S_{2^i})}\right) + 4\ln\left(\frac{2}{\delta_i}\right)\right).$$

Thus, by monotonicity of $\Delta\mathrm{VS}_{\mathcal{F},S_t}$ in t, a union bound implies that with probability at least $1-\delta/2$, for every $i \in \{0,1,\ldots,\lfloor \log_2(m) \rfloor\}$, every $t \in \{2^i,\ldots,2^{i+1}-1\}$ has

$$\Delta\mathrm{VS}_{\mathcal{F},S_t} \leq 2^{-i}\left(10\hat{n}(S_{2^i})\ln\left(\frac{e2^i}{\hat{n}(S_{2^i})}\right) + 4\ln\left(\frac{2}{\delta_i}\right)\right). \tag{3}$$

Noting that $\left\{\mathbb{1}_{\mathrm{DIS}(\mathrm{VS}_{\mathcal{F},S_{t-1}})}(x_t) - \Delta\mathrm{VS}_{\mathcal{F},S_{t-1}}\right\}_{t=1}^{\infty}$ is a martingale difference sequence with respect to $\{x_t\}_{t=1}^{\infty}$, Bernstein's inequality (for martingales) implies that with probability at least $1-\delta/2$, if (3) holds for all $i \in \{0,1,\ldots,\lfloor \log_2(m) \rfloor\}$ and $t \in \{2^i,\ldots,2^{i+1}-1\}$, then

$$\begin{aligned}\sum_{t=1}^{m} \mathbb{1}_{\mathrm{DIS}(\mathrm{VS}_{\mathcal{F},S_{t-1}})}(x_t) &\leq 1 + \sum_{i=0}^{\lfloor \log_2(m) \rfloor} \sum_{t=2^i+1}^{2^{i+1}} \mathbb{1}_{\mathrm{DIS}(\mathrm{VS}_{\mathcal{F},S_{2^i}})}(x_t) \\ &\leq \log_2\left(\frac{4}{\delta}\right) + 2e \sum_{i=0}^{\lfloor \log_2(m) \rfloor}\left(10\hat{n}(S_{2^i})\ln\left(\frac{e2^i}{\hat{n}(S_{2^i})}\right) + 4\ln\left(\frac{2}{\delta_i}\right)\right).\end{aligned}$$

Letting $\delta_i = \frac{\delta}{2\lfloor \log_2(2m) \rfloor}$, the above is at most

$$\max_{i \in \{0,1,\ldots,\lfloor \log_2(m) \rfloor\}}\left(55\hat{n}(S_{2^i})\ln\left(\frac{e2^i}{\hat{n}(S_{2^i})}\right) + 24\ln\left(\frac{4\log_2(2m)}{\delta}\right)\right)\log_2(2m).$$

■

This also implies distribution-dependent bounds on any confidence bound on the number of queries made by CAL. Specifically, let $\mathcal{B}_N(m,\delta)$ be the smallest nonnegative integer n such that $\mathbb{P}(N(m;S_m) \leq n) \geq 1-\delta$. Then the following result follows immediately from Lemma 6.

Theorem 9 *For any $m \in \mathbb{N}$ and $\delta \in (0,1)$, for any sequence δ_t in $(0,1]$ with $\sum_{i=0}^{\lfloor \log_2(m) \rfloor} \delta_{2^i} \leq \delta/2$,*

$$\begin{aligned}\max_{t \leq m} \mathcal{B}_{\hat{n}}(t,\delta) &\leq \mathcal{B}_N(m,\delta) \\ &\leq \max_{t \in \{2^i : i \in \{0,1,\ldots,\lfloor \log_2(m) \rfloor\}\}}\left(55\mathcal{B}_{\hat{n}}(t,\delta_t)\ln\left(\frac{et}{\mathcal{B}_{\hat{n}}(t,\delta_t)}\right) + 24\ln\left(\frac{8\log_2(2m)}{\delta}\right)\right)\log_2(2m).\end{aligned}$$

Proof Since Lemma 6 implies every $t \leq m$ has $\hat{n}(S_t) \leq N(m;S_m)$, we have $\mathbb{P}(\hat{n}(S_t) \leq \mathcal{B}_N(m,\delta)) \geq \mathbb{P}(N(m;S_m) \leq \mathcal{B}_N(m,\delta)) \geq 1-\delta$. Since $\mathcal{B}_{\hat{n}}(t,\delta)$ is the smallest $n \in \mathbb{N}$ with $\mathbb{P}(\hat{n}(S_t) \leq n) \geq 1-\delta$, we must therefore have $\mathcal{B}_{\hat{n}}(t,\delta) \leq \mathcal{B}_N(m,\delta)$, from which the left inequality in the claim follows by maximizing over t.

For the second inequality, the upper bound on $N(m;S_m)$ from Lemma 6 implies that, with probability at least $1-\delta/2$, $N(m;S_m)$ is at most

$$\max_{t \in \{2^i : i \in \{0,\ldots,\lfloor \log_2(m) \rfloor\}\}} \left(55\hat{n}(S_t) \ln\left(\frac{et}{\hat{n}(S_t)}\right) + 24\ln\left(\frac{8\log_2(2m)}{\delta}\right)\right)\log_2(2m).$$

Furthermore, a union bound implies that with probability at least $1-\sum_{i=0}^{\lfloor \log_2(m) \rfloor} \delta_{2^i} \geq 1-\delta/2$, every $t \in \{2^i : i \in \{0,\ldots,\lfloor \log_2(m) \rfloor\}\}$ has $\hat{n}(S_t) \leq \mathcal{B}_{\hat{n}}(t,\delta_t)$. Since $x \mapsto x\ln(et/x)$ is nondecreasing for $x \in [0,t]$, and $\mathcal{B}_{\hat{n}}(t,\delta_t) \leq t$, combining these two results via a union bound, we have that with probability at least $1-\delta$, $N(m;S_m)$ is at most

$$\max_{t \in \{2^i : i \in \{0,1,\ldots,\lfloor \log_2(m) \rfloor\}\}} \left(55\mathcal{B}_{\hat{n}}(t,\delta_t) \ln\left(\frac{et}{\mathcal{B}_{\hat{n}}(t,\delta_t)}\right) + 24\ln\left(\frac{8\log_2(2m)}{\delta}\right)\right)\log_2(2m).$$

Letting U_m denote this last quantity, note that since $N(m;S_m)$ is a nonnegative integer, $N(m;S_m) \leq U_m \Rightarrow N(m;S_m) \leq \lfloor U_m \rfloor$, so that $\mathbb{P}(N(m;S_m) \leq \lfloor U_m \rfloor) \geq 1-\delta$. Since $\mathcal{B}_N(m,\delta)$ is the *smallest* nonnegative integer n with $\mathbb{P}(N(m;S_m) \leq n) \geq 1-\delta$, we must have $\mathcal{B}_N(m,\delta) \leq \lfloor U_m \rfloor \leq U_m$. ■

In bounding the label complexity of CAL, we are primarily interested in the size of n sufficient to guarantee low error rate for every classifier in the final V_m set (since $\hat{h}$ is taken to be an arbitrary element of V_m). Specifically, we are interested in the following quantity. For $n \in \mathbb{N}$, define $M(n;S_\infty) = \min\{m \in \mathbb{N} : N(m;S_m) = n\}$ (or $M(n;S_\infty) = \infty$ if $\max_m N(m;S_m) < n$), and for any $\varepsilon,\delta \in (0,1]$, define

$$\Lambda(\varepsilon,\delta) = \min\left\{ n \in \mathbb{N} : \mathbb{P}\left(\sup_{h \in \mathrm{VS}_{\mathcal{F},S_{M(n;S_\infty)}}} \mathrm{er}(h) \leq \varepsilon \right) \geq 1-\delta \right\}.$$

Note that, for any $n \geq \Lambda(\varepsilon,\delta)$, with probability at least $1-\delta$, the classifier $\hat{h}$ produced by CAL(n) has $\mathrm{er}(\hat{h}) \leq \varepsilon$. Furthermore, for any $n < \Lambda(\varepsilon,\delta)$, with probability greater than δ, there exists a choice of $\hat{h}$ in the final step of CAL(n) for which $\mathrm{er}(\hat{h}) > \varepsilon$. Therefore, in a sense, $\Lambda(\varepsilon,\delta)$ represents the label complexity of the general family of CAL strategies (which vary only in how $\hat{h}$ is chosen from the final V_m set). We can also define an analogous quantity for passive learning by empirical risk minimization:

$$M(\varepsilon,\delta) = \min\left\{ m \in \mathbb{N} : \mathbb{P}\left(\sup_{h \in \mathrm{VS}_{\mathcal{F},S_m}} \mathrm{er}(h) \leq \varepsilon \right) \geq 1-\delta \right\}.$$

We typically expect $M(\varepsilon,\delta)$ to be larger than $\Omega(1/\varepsilon)$, and it is known $M(\varepsilon,\delta)$ is always at most $O((1/\varepsilon)(d\log(1/\varepsilon)+\log(1/\delta)))$ (e.g., Vapnik, 1998). We have the following theorem relating these two quantities.

Theorem 10 *There exists a universal constant $c \in (0,\infty)$ such that, $\forall \varepsilon, \delta \in (0,1)$, $\forall \beta \in \left(0, \frac{1-\delta}{\delta}\right)$, for any sequence δ_m in $(0,1]$ with $\sum_{i=0}^{\lfloor \log_2(M(\varepsilon,\delta/2))\rfloor} \delta_{2^i} \leq \delta/2$,*

$$\max_{m \leq M(\varepsilon,1-\beta\delta)} \mathcal{B}_{\hat{n}}(m,(1+\beta)\delta) \leq \Lambda(\varepsilon,\delta)$$
$$\leq c\left(\max_{m \leq M(\varepsilon,\delta/2)} \mathcal{B}_{\hat{n}}(m,\delta_m) \ln\left(\frac{em}{\mathcal{B}_{\hat{n}}(m,\delta_m)}\right) + \ln\left(\frac{\log_2(2M(\varepsilon,\delta/2))}{\delta}\right)\right)\log_2(2M(\varepsilon,\delta/2)).$$

Proof By definition of $M(\varepsilon,1-\beta\delta)$, $\forall m < M(\varepsilon,1-\beta\delta)$, with probability greater than $1-\beta\delta$, $\sup_{h\in \mathrm{VS}_{\mathcal{F},S_m}} \mathrm{er}(h) > \varepsilon$. Furthermore, by definition of $\mathcal{B}_{\hat{n}}(m,(1+\beta)\delta)$, $\forall n < \mathcal{B}_{\hat{n}}(m,(1+\beta)\delta)$, with probability greater than $(1+\beta)\delta$, $\hat{n}(S_m) > n$, which together with Lemma 6 implies $N(m;S_m) > n$, so that $M(n;S_\infty) < m$. Thus, fixing any $m \leq M(\varepsilon,1-\beta\delta)$ and $n < \mathcal{B}_{\hat{n}}(m,(1+\beta)\delta)$, a union bound implies that with probability exceeding δ, $M(n;S_\infty) < m$ and $\sup_{h\in \mathrm{VS}_{\mathcal{F},S_{m-1}}} \mathrm{er}(h) > \varepsilon$. By monotonicity of $t \mapsto \mathrm{VS}_{\mathcal{F},S_t}$, this implies that with probability greater than δ, $\sup_{h\in \mathrm{VS}_{\mathcal{F},S_{M(n;S_\infty)}}} \mathrm{er}(h) > \varepsilon$, so that $\Lambda(\varepsilon,\delta) > n$.

For the upper bound, Lemma 6 and a union bound imply that, with probability at least $1-\delta/2$,

$$N(M(\varepsilon,\delta/2);S_{M(\varepsilon,\delta/2)}) \leq$$
$$c'\left(\max_{m \leq M(\varepsilon,\delta/2)} \mathcal{B}_{\hat{n}}(m,\delta_m) \ln\left(\frac{em}{\mathcal{B}_{\hat{n}}(m,\delta_m)}\right) + \ln\left(\frac{\log_2(2M(\varepsilon,\delta/2))}{\delta}\right)\right)\log_2(2M(\varepsilon,\delta/2)),$$

for a universal constant $c' > 0$. In particular, this implies that for any n at least this large, with probability at least $1-\delta/2$, $M(n+1;S_\infty) \geq M(\varepsilon,\delta/2)$. Furthermore, by definition of $M(\varepsilon,\delta/2)$ and monotonicity of $m \mapsto \sup_{h\in \mathrm{VS}_{\mathcal{F},S_m}} \mathrm{er}(h)$, with probability at least $1-\delta/2$, every $m \geq M(\varepsilon,\delta/2)$ has $\sup_{h\in \mathrm{VS}_{\mathcal{F},S_m}} \mathrm{er}(h) \leq \varepsilon$. By a union bound, with probability at least $1-\delta$, $\sup_{h\in \mathrm{VS}_{\mathcal{F},S_{M(n+1;S_\infty)}}} \mathrm{er}(h) \leq \varepsilon$. This implies $\Lambda(\varepsilon,\delta) \leq n+1$, so that the result holds (for instance, it suffices to take $c = c'+2$). ■

For instance, $\delta_m = \delta/\left(2\log_2(2M(\varepsilon,\delta/2))\right)$ might be a natural choice in the above result.

Another implication of these results is a complement to Theorem 4 that was presented in Theorem 5 above.

Proof of Theorem 5 Lemma 29 in Appendix A and monotonicity of $\varepsilon \mapsto \theta(\varepsilon)$ imply that, for $m = \lceil 1/r_0 \rceil$,

$$\mathcal{B}_N(m,\delta) \leq 8 \vee c_0\theta(dr_0/2)\left(d\ln(e\theta(dr_0/2)) + \ln\left(\frac{\log_2(2/r_0)}{\delta}\right)\right)\log_2\left(\frac{2}{r_0}\right)$$
$$\leq (c_0 \vee 8)\theta(dr_0/2)\left(d\ln(e\theta(dr_0/2)) + \ln\left(\frac{\log_2(2/r_0)}{\delta}\right)\right)\log_2\left(\frac{2}{r_0}\right),$$

for a finite universal constant $c_0 > 0$. The result then follows from Theorem 9 and the fact that $\theta(dr_0/2) \leq 2\theta(dr_0)$ (Hanneke, 2014). ■

This also implies the following corollary on the necessary and sufficient conditions for CAL to provide exponential improvements in label complexity when passive learning by empirical risk minimization has $\Omega(1/\varepsilon)$ sample complexity (which is typically the case).[6]

6. All of these equivalences continue to hold even when this $M(\varepsilon,\cdot) = \Omega(1/\varepsilon)$ condition fails, excluding statements 1 and 2, which would then be implied by the others but not vice versa.

Corollary 11 (Characterization of CAL) *If $d < \infty$, and $\exists \delta_0 \in (0,1)$ such that $M(\varepsilon, \delta_0) = \Omega(1/\varepsilon)$, then the following are all equivalent:*

1. $\Lambda(\varepsilon, \delta) = O\left(\text{polylog}\left(\frac{1}{\varepsilon}\right)\log\left(\frac{1}{\delta}\right)\right)$,
2. $\Lambda\left(\varepsilon, \frac{1}{40}\right) = O\left(\text{polylog}\left(\frac{1}{\varepsilon}\right)\right)$,
3. $\mathcal{B}_{\hat{n}}(m, \delta) = O\left(\text{polylog}(m)\log\left(\frac{1}{\delta}\right)\right)$,
4. $\mathcal{B}_{\hat{n}}\left(m, \frac{1}{20}\right) = O\left(\text{polylog}(m)\right)$,
5. $\theta(r_0) = O\left(\text{polylog}\left(\frac{1}{r_0}\right)\right)$,
6. $\mathcal{B}_{\Delta}(m, \delta) = O\left(\frac{\text{polylog}(m)}{m}\log\left(\frac{1}{\delta}\right)\right)$,
7. $\mathcal{B}_{\Delta}\left(m, \frac{1}{9}\right) = O\left(\frac{\text{polylog}(m)}{m}\right)$,
8. $\mathcal{B}_{N}(m, \delta) = O\left(\text{polylog}(m)\log\left(\frac{1}{\delta}\right)\right)$,
9. $\mathcal{B}_{N}\left(m, \frac{1}{20}\right) = O\left(\text{polylog}(m)\right)$,

where $\mathcal{F}$ and P are considered constant, so that the big-O hides $(\mathcal{F}, P)$-dependent constant factors here (but no factors depending on ε, δ, m, or r_0).[7]

Proof We decompose the proof into a series of implications. Specifically, we show that $3 \Rightarrow 4 \Rightarrow 5 \Rightarrow 8 \Rightarrow 3$, $8 \Rightarrow 9 \Rightarrow 4$, $5 \Rightarrow 1 \Rightarrow 2 \Rightarrow 4$, and $3 \Rightarrow 6 \Rightarrow 7 \Rightarrow 5$. These implications form a strongly connected directed graph, and therefore establish equivalence of the statements.

(3 $\Rightarrow$ 4) If $\mathcal{B}_{\hat{n}}(m, \delta) = O\left(\text{polylog}(m)\log\left(\frac{1}{\delta}\right)\right)$, then in particular there is some (sufficiently small) constant $\delta_1 \in (0, 1/20)$ for which $\mathcal{B}_{\hat{n}}(m, \delta_1) = O\left(\text{polylog}(m)\right)$, and since $\delta \mapsto \mathcal{B}_{\hat{n}}(m, \delta)$ is nonincreasing, $\mathcal{B}_{\hat{n}}\left(m, \frac{1}{20}\right) \leq \mathcal{B}_{\hat{n}}(m, \delta_1)$, so that $\mathcal{B}_{\hat{n}}\left(m, \frac{1}{20}\right) = O\left(\text{polylog}(m)\right)$ as well.

(4 $\Rightarrow$ 5) If $\mathcal{B}_{\hat{n}}\left(m, \frac{1}{20}\right) = O\left(\text{polylog}(m)\right)$, then

$$\max_{m \leq 1/r_0} \mathcal{B}_{\hat{n}}\left(m, \frac{1}{20}\right) = O\left(\max_{m \leq 1/r_0} \text{polylog}(m)\right) = O\left(\text{polylog}\left(\frac{1}{r_0}\right)\right).$$

Therefore, Theorem 4 implies

$$\theta(r_0) \leq \max\left\{\max_{m \leq \lceil 1/r_0 \rceil} 16\mathcal{B}_{\hat{n}}\left(m, \frac{1}{20}\right), 512\right\}$$
$$\leq 528 + 16 \max_{m \leq 1/r_0} \mathcal{B}_{\hat{n}}\left(m, \frac{1}{20}\right) = O\left(\text{polylog}\left(\frac{1}{r_0}\right)\right).$$

7. In fact, we may choose freely whether or not to allow the big-O to hide f^*-dependent constants, or P-dependent constants in general, as long as the *same* interpretation is used for all of these statements. Though validity for each of these interpretations generally does not imply validity for the others, the proof remains valid regardless of which of these interpretations we choose, as long as we stick to the same interpretation throughout the proof.

(5 ⇒ 8) If $\theta(r_0) = O\left(\text{polylog}\left(\frac{1}{r_0}\right)\right)$, then Lemma 29 in Appendix A implies that $\mathcal{B}_N(m,\delta) = O\left(\text{polylog}(m)\log\left(\frac{1}{\delta}\right)\right)$.

(8 ⇒ 3) If $\mathcal{B}_N(m,\delta) = O\left(\text{polylog}(m)\log\left(\frac{1}{\delta}\right)\right)$, then Theorem 9 implies

$$\mathcal{B}_{\hat{n}}(m,\delta) \leq \mathcal{B}_N(m,\delta) = O\left(\text{polylog}(m)\log\left(\frac{1}{\delta}\right)\right).$$

(8 ⇒ 9) If $\mathcal{B}_N(m,\delta) = O\left(\text{polylog}(m)\log\left(\frac{1}{\delta}\right)\right)$, then for any sufficiently small value $\delta_2 \in (0,1/20)$, $\mathcal{B}_N(m,\delta_2) = O(\text{polylog}(m))$; monotonicity of $\delta \mapsto \mathcal{B}_N(m,\delta)$ further implies $\mathcal{B}_N\left(m,\frac{1}{20}\right) \leq \mathcal{B}_N(m,\delta_2)$, so that $\mathcal{B}_N\left(m,\frac{1}{20}\right) = O(\text{polylog}(m))$.

(9 ⇒ 4) When $\mathcal{B}_N\left(m,\frac{1}{20}\right) = O(\text{polylog}(m))$, Theorem 9 implies that $\mathcal{B}_{\hat{n}}\left(m,\frac{1}{20}\right) \leq \mathcal{B}_N\left(m,\frac{1}{20}\right) = O\left(\text{polylog}(m)\right)$.

(5 ⇒ 1) If $\theta(r_0) = O\left(\text{polylog}\left(\frac{1}{r_0}\right)\right)$, then Lemma 30 in Appendix A implies that $\Lambda(\varepsilon,\delta) = O\left(\text{polylog}\left(\frac{1}{\varepsilon}\right)\log\left(\frac{1}{\delta}\right)\right)$.

(1 ⇒ 2) If $\Lambda(\varepsilon,\delta) = O\left(\text{polylog}\left(\frac{1}{\varepsilon}\right)\log\left(\frac{1}{\delta}\right)\right)$, then for any sufficiently small value $\delta_3 \in (0,1/40]$, $\Lambda(\varepsilon,\delta_3) = O\left(\text{polylog}\left(\frac{1}{\varepsilon}\right)\right)$; furthermore, monotonicity of $\delta \mapsto \Lambda(\varepsilon,\delta)$ implies $\Lambda\left(\varepsilon,\frac{1}{40}\right) \leq \Lambda(\varepsilon,\delta_3)$, so that $\Lambda\left(\varepsilon,\frac{1}{40}\right) = O\left(\text{polylog}\left(\frac{1}{\varepsilon}\right)\right)$ as well.

(2 ⇒ 4) Let $c \in (0,1]$ and $\varepsilon_0 \in (0,1)$ be constants such that, $\forall \varepsilon \in (0,\varepsilon_0)$, $M(\varepsilon,\delta_0) \geq \frac{c}{\varepsilon}$. For any $\delta \in (0,1/20)$, if $\frac{19}{20}+\delta \leq \delta_0$, then $M\left(\varepsilon,\frac{19}{20}+\delta\right) \geq M(\varepsilon,\delta_0) \geq c/\varepsilon$; otherwise, if $\frac{19}{20}+\delta > \delta_0$, then letting $m = M(\varepsilon,\frac{19}{20}+\delta)$ and $\mathcal{L}_i = \{(x_{m(i-1)+1},y_{m(i-1)+1}),\ldots,(x_{mi},y_{mi})\}$ for $i \in \mathbb{N}$, we have that $\forall k \in \mathbb{N}$,

$$\begin{aligned}\mathbb{P}\left(\sup_{h\in \text{VS}_{\mathcal{F},S_{mk}}} \text{er}(h) > \varepsilon\right) &\leq \mathbb{P}\left(\min_{i\leq k}\sup_{h\in \text{VS}_{\mathcal{F},\mathcal{L}_i}} \text{er}(h) > \varepsilon\right) \\ &= \prod_{i=1}^{k}\mathbb{P}\left(\sup_{h\in \text{VS}_{\mathcal{F},\mathcal{L}_i}} \text{er}(h) > \varepsilon\right) \leq \left(\frac{19}{20}+\delta\right)^k,\end{aligned}$$

so that setting $k = \left\lceil \frac{\ln(1/\delta_0)}{\ln(1/(\frac{19}{20}+\delta))} \right\rceil$ reveals that

$$M(\varepsilon,\delta_0) \leq M\left(\varepsilon,\frac{19}{20}+\delta\right)\left\lceil \frac{\ln(1/\delta_0)}{\ln(1/(\frac{19}{20}+\delta))} \right\rceil. \tag{4}$$

Since $\ln(x) < x-1$ for $x \in (0,1)$, we have $\ln(1/(\frac{19}{20}+\delta)) = -\ln(\frac{19}{20}+\delta) > -(\frac{19}{20}+\delta-1) = \frac{1}{20}-\delta$; together with the fact that $\frac{1}{20}-\delta < 1$, this implies

$$\begin{aligned}\left\lceil \frac{\ln(1/\delta_0)}{\ln(1/(\frac{19}{20}+\delta))} \right\rceil &\leq \left\lceil \frac{\ln(1/\delta_0)}{\frac{1}{20}-\delta} \right\rceil < \frac{\ln(1/\delta_0)}{\frac{1}{20}-\delta}+1 \\ &< \frac{\ln(1/\delta_0)}{\frac{1}{20}-\delta}+\frac{1}{\frac{1}{20}-\delta} = \frac{\ln(e/\delta_0)}{\frac{1}{20}-\delta}.\end{aligned}$$

Plugging this into (4) reveals that

$$M\left(\varepsilon, \frac{19}{20}+\delta\right) \geq \frac{\frac{1}{20}-\delta}{\ln(e/\delta_0)} M(\varepsilon, \delta_0) \geq \frac{c(\frac{1}{20}-\delta)}{\ln(e/\delta_0)} \frac{1}{\varepsilon}.$$

If $\Lambda\left(\varepsilon, \frac{1}{40}\right) = O\left(\text{polylog}\left(\frac{1}{\varepsilon}\right)\right)$, then Theorem 10 (with $\beta = \frac{1}{20\delta} - 1$ and $\delta = 1/40$) implies

$$\max_{t \leq \frac{c/40}{\ln(e/\delta_0)} \frac{1}{\varepsilon}} \mathcal{B}_{\hat{n}}\left(t, \frac{1}{20}\right) \leq \Lambda\left(\varepsilon, \frac{1}{40}\right) = O\left(\text{polylog}\left(\frac{1}{\varepsilon}\right)\right).$$

This implies that, $\forall m \in \mathbb{N}$,

$$\begin{aligned}\mathcal{B}_{\hat{n}}\left(m, \frac{1}{20}\right) &\leq \Lambda\left(\frac{c/40}{m\ln(e/\delta_0)}, \frac{1}{40}\right) \\ &= O\left(\text{polylog}\left(\frac{m\ln(e/\delta_0)}{(c/40)}\right)\right) = O\left(\text{polylog}(m)\right).\end{aligned}$$

(3 $\Rightarrow$ 6) Lemma 8 implies that with probability at least $1-\delta/2$,

$$\Delta \text{VS}_{\mathcal{F}, S_m} \leq \frac{1}{m}\left(10\hat{n}(S_m)\ln\left(\frac{em}{\hat{n}(S_m)}\right) + 4\ln\left(\frac{4}{\delta}\right)\right),$$

while the definition of $\mathcal{B}_{\hat{n}}\left(m, \frac{\delta}{2}\right)$ implies that $\hat{n}(S_m) \leq \mathcal{B}_{\hat{n}}\left(m, \frac{\delta}{2}\right)$ with probability at least $1-\delta/2$. By a union bound, both of these occur with probability at least $1-\delta$; together with the facts that $x \mapsto x\ln(em/x)$ is nondecreasing on $(0, m]$ and $\mathcal{B}_{\hat{n}}\left(m, \frac{\delta}{2}\right) \leq m$, this implies

$$\begin{aligned}\mathcal{B}_{\Delta}(m, \delta) &\leq \frac{1}{m}\left(10\mathcal{B}_{\hat{n}}\left(m, \frac{\delta}{2}\right)\ln\left(\frac{em}{\mathcal{B}_{\hat{n}}\left(m, \frac{\delta}{2}\right)}\right) + 4\ln\left(\frac{4}{\delta}\right)\right) \\ &= O\left(\frac{1}{m}\left(\mathcal{B}_{\hat{n}}\left(m, \frac{\delta}{2}\right)\log(m) + \log\left(\frac{1}{\delta}\right)\right)\right).\end{aligned}$$

Thus, if $\mathcal{B}_{\hat{n}}(m, \delta) = O\left(\text{polylog}(m)\log\left(\frac{1}{\delta}\right)\right)$, then we have

$$\mathcal{B}_{\Delta}(m, \delta) = O\left(\frac{\text{polylog}(m)}{m}\log\left(\frac{1}{\delta}\right)\right).$$

(6 $\Rightarrow$ 7) If $\mathcal{B}_{\Delta}(m, \delta) = O\left(\frac{\text{polylog}(m)}{m}\log\left(\frac{1}{\delta}\right)\right)$, then there exists a sufficiently small constant $\delta_4 \in (0, 1/9]$ such that $\mathcal{B}_{\Delta}(m, \delta_4) = O\left(\frac{\text{polylog}(m)}{m}\right)$; in fact, combined with monotonicity of $\delta \mapsto \mathcal{B}_{\Delta}(m, \delta)$, this implies $\mathcal{B}_{\Delta}\left(m, \frac{1}{9}\right) = O\left(\frac{\text{polylog}(m)}{m}\right)$ as well.

(7 ⇒ 5) If $\mathcal{B}_\Delta\left(m, \frac{1}{9}\right) = O\left(\frac{\text{polylog}(m)}{m}\right)$, then Lemma 31 in Appendix A implies

$$\begin{aligned}\theta(r_0) &\leq \max\left\{\sup_{r\in(r_0,1/2)} \frac{7\mathcal{B}_\Delta\left(\lfloor 1/r \rfloor, \frac{1}{9}\right)}{r}, 2\right\} \\ &\leq 2 + 14 \max_{m\leq 1/r_0} m\mathcal{B}_\Delta\left(m, \frac{1}{9}\right) \\ &= O\left(\max_{m\leq 1/r_0} \text{polylog}(m)\right) = O\left(\text{polylog}\left(\frac{1}{r_0}\right)\right).\end{aligned}$$

■

5. Applications

In this section, we state bounds on the complexity measures studied above, for various hypothesis classes $\mathcal{F}$ and distributions P, which can then be used in conjunction with the above results. In each case, combining the result with theorems above yields a bound on the label complexity of CAL that is smaller than the best known result in the published literature for that problem.

5.1 Linear Separators under Mixtures of Gaussians

The first result, due to El-Yaniv and Wiener (2010), applies to the problem of learning linear separators under a mixture of Gaussians distribution. Specifically, for $k \in \mathbb{N}$, the class of linear separators in $\mathbb{R}^k$ is defined as the set of classifiers $(x_1, \ldots, x_k) \mapsto \text{sign}(b + \sum_{i=1}^k x_i w_i)$, where the values $b, w_1, \ldots, w_k \in \mathbb{R}$ are free parameters specifying the classifier, with $\sum_{i=1}^k w_i^2 = 1$, and where $\text{sign}(t) = 2\mathbb{1}_{[0,\infty)}(t) - 1$. In this work, we also include the two constant functions $x \mapsto -1$ and $x \mapsto +1$ as members of the class of linear separators.

Theorem 12 (El-Yaniv and Wiener, 2010, Lemma 32) *For $t, k \in \mathbb{N}$, there is a finite constant $c_{k,t} > 0$ such that, for $\mathcal{F}$ the space of linear separators on $\mathbb{R}^k$, and for P with marginal distribution over X that is a mixture of t multivariate normal distributions with diagonal covariance matrices of full rank, $\forall m \geq 2$,*

$$\mathcal{B}_{\hat{n}}\left(m, \frac{1}{20}\right) \leq c_{k,t}(\log(m))^{k-1}.$$

Combining this result with Theorem 4 implies that there is a constant $c_{k,t} \in (0,\infty)$ such that, for $\mathcal{F}$ and P as in Theorem 12, $\forall r_0 \in (0, 1/2]$,

$$\theta(r_0) \leq c_{k,t}\left(\log\left(\frac{1}{r_0}\right)\right)^{k-1}.$$

In particular, plugging this into the label complexity bound of Hanneke (2011) for CAL (Lemma 30 of Appendix A) yields the following bound on the label complexity of CAL, which has an improved asymptotic dependence on ε compared to the previous best known result, due to El-Yaniv and Wiener (2012), reducing the exponent on the logarithmic factor from $\Theta(k^2)$ to $\Theta(k)$, and reducing the dependence on δ from $\text{poly}(1/\delta)$ to $\log(1/\delta)$.

Corollary 13 *For $t,k \in \mathbb{N}$, there is a finite constant $c_{k,t} > 0$ such that, for $\mathcal{F}$ the space of linear separators on $\mathbb{R}^k$, and for P with marginal distribution over $\mathcal{X}$ that is a mixture of t multivariate normal distributions with diagonal covariance matrices of full rank, $\forall \varepsilon, \delta \in (0,1/2]$,*

$$\Lambda(\varepsilon,\delta) \leq c_{k,t}\left(\log\left(\frac{1}{\varepsilon}\right)\right)^k \log\left(\frac{\log(1/\varepsilon)}{\delta}\right).$$

Corollary 13 is particularly interesting in light of a lower bound of El-Yaniv and Wiener (2012) for this problem, showing that there exists a distribution P of the type described in Corollary 13 for which $\mathcal{B}_N(m,\delta) = \Omega\left((\log(m))^{\frac{k-1}{2}}\right)$.

5.2 Axis-aligned Rectangles under Product Densities

The next result applies to the problem of learning axis-aligned rectangles under product densities over $\mathbb{R}^k$: that is, classifiers $h((x'_1,\ldots,x'_k)) = 2\prod_{j=1}^k \mathbb{1}_{[a_j,b_j]}(x'_j) - 1$, for values $a_1,\ldots,a_k,b_1,\ldots,b_k \in \mathbb{R}$. The result specifically applies to rectangles with a probability at least $\lambda > 0$ of classifying a random point positive. This result represents a refinement of a result of Hanneke (2007b): specifically, reducing a factor of k^2 to a factor of k.

Theorem 14 *For $k,m \in \mathbb{N}$ and $\lambda,\delta \in (0,1)$, for any P with marginal distribution over $\mathcal{X}$ that is a product distribution with marginals having continuous CDFs, and for $\mathcal{F}$ the space of axis-aligned rectangles h on $\mathbb{R}^k$ with $P((x,y) : h(x) = 1) \geq \lambda$,*

$$\mathcal{B}_{\hat{n}}(m,\delta) \leq \frac{8k}{\lambda}\ln\left(\frac{8k}{\delta}\right).$$

Proof The proof is based on a slight refinement of an argument of Hanneke (2007b). For $(X,Y) \sim P$, denote $(X_1,\ldots,X_k) \triangleq X$, let G_i be the CDF of X_i, and define $G(X_1,\ldots,X_k) \triangleq (G_1(X_1),\ldots,G_k(X_k))$. Then the random variable $X' \triangleq (X'_1,\ldots,X'_k) \triangleq (G_1(X_1),\ldots,G_k(X_k)) = G(X)$ is uniform in $(0,1)^k$; to see this, note that since $X_1,\ldots,X_k$ are independent, so are $G_1(X_1),\ldots,G_k(X_k)$, and that for each $i \leq k$, $\forall t \in (0,1)$, $\mathbb{P}(G_i(X_i) \leq t) = \sup_{x\in\mathbb{R}:G_i(x)=t}\mathbb{P}(X_i \leq x) = \sup_{x\in\mathbb{R}:G_i(x)=t} G_i(x) = t$, where the first equality is by monotonicity and continuity of G_i and the intermediate value theorem (since $\lim_{x\to-\infty} G_i(x) = 0 < t$ and $\lim_{x\to\infty} G_i(x) = 1 > t$), and the second equality is by definition of G_i. Fix any $h \in \mathcal{F}$, let $a_1,\ldots,a_k,b_1,\ldots,b_k \in \mathbb{R}$ be the values such that $h((z_1,\ldots,z_k)) = 2\prod_{i=1}^k \mathbb{1}_{[a_i,b_i]}(z_i) - 1$ for all $(z_1,\ldots,z_k) \in \mathbb{R}^k$, and define $H_h((z_1,\ldots,z_k)) = 2\prod_{i=1}^k \mathbb{1}_{[G_i(a_i),G_i(b_i)]}(z_i) - 1$. Clearly H_h is an axis-aligned rectangle. Furthermore, for every $z \in \mathbb{R}^k$ with $h(z) = +1$, monotonicity of the G_i functions implies $H_h(G(z)) = +1$ as well. Therefore, $\mathbb{P}(H_h(X') = +1) \geq \mathbb{P}(h(X) = +1) \geq \lambda$.

Let $G_i^{-1}(t) = \min\{s : G_i(s) = t\}$ for $t \in (0,1)$, which is well-defined by continuity of G_i and the intermediate value theorem, combined with the facts that $\lim_{z\to\infty} G_i(z) = 1$ and $\lim_{z\to-\infty} G_i(z) = 0$. Let T_i denote the set of discontinuity points of G_i^{-1} in $(0,1)$. Fix any $(z_1,\ldots,z_k) \in \mathbb{R}^k$ with $h((z_1,\ldots,z_k)) = -1$ and $G(z_1,\ldots,z_k) \in (0,1)^k$. In particular, this implies $\exists i \in \{1,\ldots,k\}$ such that $z_i \notin [a_i,b_i]$. For this i, we have $G_i(z_i) \notin (G_i(a_i),G_i(b_i))$ by monotonicity of G_i. Therefore, if $H_h(G(z_1,\ldots,z_k)) = +1$, we must have either $z_i < a_i$ and $G_i(z_i) = G_i(a_i)$, or $z_i > b_i$ and $G_i(z_i) = G_i(b_i)$. In the former case, for any ε with $0 < \varepsilon < 1 - G_i(z_i)$, $G_i^{-1}(G_i(z_i)+\varepsilon) = G_i^{-1}(G_i(a_i)+\varepsilon) > a_i$, while $G_i^{-1}(G_i(z_i)) \leq z_i$, and since $z_i < a_i$, we must have $G_i(z_i) \in T_i$. Similarly, in the latter case ($z_i > b_i$ and $G_i(z_i) = G_i(b_i)$), any ε with $0 < \varepsilon < 1 - G_i(z_i)$ has $G_i^{-1}(G_i(b_i)+\varepsilon) = G_i^{-1}(G_i(z_i)+\varepsilon) > z_i$,

while $G_i^{-1}(G_i(b_i)) \leq b_i$, and since $z_i > b_i$, we have $G_i(b_i) \in T_i$; since $G_i(z_i) = G_i(b_i)$, this also implies $G_i(z_i) \in T_i$. Thus, any $(z_1,\ldots,z_k) \in \mathbb{R}^k$ with $H_h(G(z_1,\ldots,z_k)) \neq h((z_1,\ldots,z_k))$ must have some $i \in \{1,\ldots,k\}$ with $G_i(z_i) \in T_i$.

For each $i \in \{1,\ldots,k\}$, since G_i is nondecreasing, G_i^{-1} is also nondecreasing, and this implies G_i^{-1} has at most countably many discontinuity points (see e.g., Kolmogorov and Fomin, 1975, Section 31, Theorem 1). Furthermore, for every $t \in \mathbb{R}$,

$$\begin{aligned}\mathbb{P}(G_i(X_i) = t) &\leq \mathbb{P}\left(\inf\{x \in \mathbb{R} : G_i(x) = t\} \leq X_i \leq \sup\{x \in \mathbb{R} : G_i(x) = t\}\right) \\ &= G_i(\sup\{x \in \mathbb{R} : G_i(x) = t\}) - G_i(\inf\{x \in \mathbb{R} : G_i(x) = t\}) = t - t = 0,\end{aligned}$$

where the inequality is due to monotonicity of G_i, the first equality is by definition of G_i as the CDF and by continuity of G_i (which implies $\mathbb{P}(X_i < x) = G_i(x)$), and the second equality is due to continuity of G_i. Therefore,

$$\mathbb{P}\left(\exists h \in \mathcal{F} : H_h(G(X)) \neq h(X)\right) \leq \mathbb{P}\left(\exists i \in \{1,\ldots,k\} : G_i(X_i) \in T_i\right) \leq \sum_{i=1}^{k} \sum_{t \in T_i} \mathbb{P}(G_i(X_i) = t) = 0.$$

By a union bound, this implies that with probability 1, for every $h \in \mathcal{F}$, every $(x,y) \in S_m$ has $H_h(G(x)) = h(x)$. In particular, we have that with probability 1, every classification of the sequence $\{x_1,\ldots,x_m\}$ realized by classifiers in $\mathcal{F}$ is also realized as a classification of the i.i.d. Uniform$((0,1)^k)$ sequence $\{G(x_1),\ldots,G(x_m)\}$ by the set $\mathcal{F}'$ of axis-aligned rectangles h' with $\mathbb{P}(h'(X') = +1) \geq \lambda$. This implies that $\mathcal{B}_{\hat{n}}(m,\delta) \leq \min\{b \in \mathbb{N} \cup \{0\} : \mathbb{P}(\hat{n}(\mathcal{F}', \{(G(x),y) : (x,y) \in S_m\}) \leq b) \geq 1-\delta\}$ (in fact, one can show they are equal). Therefore, since the right hand side is the value of $\mathcal{B}_{\hat{n}}(m,\delta)$ one would get from the case of P having marginal $P(\cdot \times \mathcal{Y})$ over $\mathcal{X}$ that is Uniform$((0,1)^k)$, without loss of generality, it suffices to bound $\mathcal{B}_{\hat{n}}(m,\delta)$ for this special case. Toward this end, for the remainder of this proof, we assume P has marginal $P(\cdot \times \mathcal{Y})$ over $\mathcal{X}$ uniform in $(0,1)^k$.

Let $m \in \mathbb{N}$, and let $\mathcal{U} = \{x_1,\ldots,x_m\}$, the unlabeled portion of the first m data points. Further denote by $\mathcal{U}^+ = \{x_i \in \mathcal{U} : f^*(x_i) = +1\}$, and $\mathcal{U}^- = \mathcal{U} \setminus \mathcal{U}^+$. For each $i \in \mathbb{N}$, express x_i explicitly in vector form as $(x_{i1},\ldots,x_{ik})$. If $\mathcal{U}^+ \neq \emptyset$, for each $j \in \{1,\ldots,k\}$, let $a_j = \min\{x_{ij} : x_i \in \mathcal{U}^+\}$ and $b_j = \max\{x_{ij} : x_i \in \mathcal{U}^+\}$. Denote by $h_{\text{clos}}(x) = 2\mathbb{1}_{\times_{j=1}^k [a_j,b_j]}(x) - 1$, the *closure* hypothesis; for completeness, when $\mathcal{U}^+ = \emptyset$, let $h_{\text{clos}}(x) = -1$ for all x.

First, note that if $m < \frac{2e}{\lambda}\left(2k + \ln\left(\frac{2}{\delta}\right)\right)$, the result trivially holds, since $\hat{n}(S_m) \leq m$ always, and $\frac{2e}{\lambda}\left(2k + \ln\left(\frac{2}{\delta}\right)\right) \leq \frac{8k}{\lambda}\ln\left(\frac{8k}{\delta}\right)$. Otherwise, if $m \geq \frac{2e}{\lambda}\left(2k + \ln\left(\frac{2}{\delta}\right)\right)$, a result of Auer and Ortner (2004) implies that, on an event E_{clos} of probability at least $1 - \delta/2$, $P((x,y) : h_{\text{clos}}(x) \neq f^*(x)) \leq \lambda/2$. In particular, since $P((x,y) : f^*(x) = +1) \geq \lambda$, on this event we must have $P((x,y) : h_{\text{clos}}(x) = +1) \geq \lambda/2$. Furthermore, this implies $\mathcal{U}^+ \neq \emptyset$ on E_{clos}.

Now fix any $j \in \{1,\ldots,k\}$. Let $x_j^{(aj)}$ denote the value x_{ij} for the point $x_i \in \mathcal{U}$ with largest x_{ij} such that $x_{ij} < a_j$, and for all $j' \neq j$, $x_{ij'} \in [a_{j'}, b_{j'}]$; if no such point exists, let $x_j^{(aj)} = 0$. Let $\mathcal{U}^{(aj)} = \{x_i \in \mathcal{U} : x_{ij} < a_j\}$. Let $m^{(aj)} = |\mathcal{U}^{(aj)}|$, and enumerate the points in $\mathcal{U}^{(aj)}$ in decreasing order of x_{ij}, so that $i_1,\ldots,i_{m^{(aj)}}$ are distinct indices such that each $t \in \{1,\ldots,m^{(aj)}\}$ has $x_{i_t} \in \mathcal{U}^{(aj)}$, and each $t \in \{1,\ldots,m^{(aj)}-1\}$ has $x_{i_{t+1}j} \leq x_{i_t j}$. Since $P((x,y) : h_{\text{clos}}(x) = +1) \geq \lambda/2$ on E_{clos}, it must be that the volume of $\times_{j' \neq j}[a_{j'}, b_{j'}]$ is at least $\lambda/2$. Therefore, working under the conditional distribution given $\mathcal{U}^+$ and $m^{(aj)}$, on E_{clos}, for each $t \in \{1,\ldots,m^{(aj)}\}$, with conditional probability at least $\lambda/2$, we have $\forall j' \neq j$, $x_{i_t j'} \in [a_{j'}, b_{j'}]$. Therefore, the value $t^{(aj)} \triangleq \min\{t : \forall j' \neq j, x_{i_t j'} \in$

$[a_{j'}, b_{j'}]\} \cup \{m^{(aj)}\}$ is bounded by a Geometric random variable with parameter $\lambda/2$. In particular, this implies that with conditional probability at least $1 - \frac{\delta}{4k}$, $t^{(aj)} \leq \left\lceil \frac{2}{\lambda} \ln\left(\frac{4k}{\delta}\right) \right\rceil$. Letting $A^{(aj)} = \{x_i \in \mathcal{U} : x_j^{(aj)} \leq x_{ij} < a_j\}$, we note that $|A^{(aj)}| \leq t^{(aj)}$ with probability 1, so that the above reasoning, combined with the law of total probability, implies that there is an event $E^{(aj)}$ of probability at least $1 - \frac{\delta}{4k}$ such that, on $E^{(aj)} \cap E_{\text{clos}}$, $|A^{(aj)}| \leq \left\lceil \frac{2}{\lambda} \ln\left(\frac{4k}{\delta}\right) \right\rceil$. For the symmetric case, define $x_j^{(bj)}$ as the value x_{ij} for the point $x_i \in \mathcal{U}$ with smallest x_{ij} such that $x_{ij} > b_j$, and for all $j' \neq j$, $x_{ij'} \in [a_{j'}, b_{j'}]$; if no such point x_i exists, define $x_j^{(bj)} = 1$. Define $A^{(bj)} = \{x_i \in \mathcal{U} : b_j < x_{ij} \leq x_j^{(bj)}\}$. By the same reasoning as above, there is an event $E^{(bj)}$ of probability at least $1 - \frac{\delta}{4k}$ such that, on $E^{(bj)} \cap E_{\text{clos}}$, $|A^{(bj)}| \leq \left\lceil \frac{2}{\lambda} \ln\left(\frac{4k}{\delta}\right) \right\rceil$. Applying this to all values of j, and letting $A = \bigcup_{j=1}^{k} A^{(aj)} \cup A^{(bj)}$, we have that on the event $E_{\text{clos}} \cap \bigcap_{j=1}^{k} E^{(aj)} \cap E^{(bj)}$,

$$|A| \leq 2k \left\lceil \frac{2}{\lambda} \ln\left(\frac{4k}{\delta}\right) \right\rceil.$$

Furthermore, a union bound implies that the event $E_{\text{clos}} \cap \bigcap_{j=1}^{k} E^{(aj)} \cap E^{(bj)}$ has probability at least $1 - \delta$. For the remainder of the proof, we suppose this event occurs.

Next, let $B = \left\{ \operatorname*{argmin}_{x_i \in \mathcal{U}^+} x_{ij} : j \in \{1, \ldots, k\} \right\} \cup \left\{ \operatorname*{argmax}_{x_i \in \mathcal{U}^+} x_{ij} : j \in \{1, \ldots, k\} \right\}$, and note that $|B| \leq 2k$. Finally, we conclude the proof by showing that the set $A \cup B$ has the property that $\{h \in \mathcal{F} : \forall x \in A \cup B, h(x) = f^*(x)\} = \text{VS}_{\mathcal{F}, S_m}$, which implies $\{(x_i, y_i) : x_i \in A \cup B\}$ is a version space compression set, so that $\hat{n}(S_m) \leq |A \cup B|$, and hence $\mathcal{B}_{\hat{n}}(m, \delta) \leq 2k + 2k \left\lceil \frac{2}{\lambda} \ln\left(\frac{4k}{\delta}\right) \right\rceil \leq \frac{8k}{\lambda} \ln\left(\frac{4k}{\delta}\right)$. To prove that $A \cup B$ has this property, first note that any $h \in \mathcal{F}$ with $h(x_i) = +1$ for all $x_i \in B$, must have $\mathcal{U}^+ \supseteq \{x_i \in \mathcal{U}^+ : h(x_i) = +1\} \supseteq \mathcal{U}^+ \cap \times_{j=1}^{k} [\min_{x_i \in \mathcal{U}^+} x_{ij}, \max_{x_i \in \mathcal{U}^+} x_{ij}] = \mathcal{U}^+$, so that $\{x_i \in \mathcal{U} : h(x_i) = +1\} \supseteq \mathcal{U}^+ = \{x_i \in \mathcal{U} : f^*(x_i) = +1\}$. Next, for any $x_i \in \mathcal{U}^- \setminus (A \cup B)$, $\exists j \in \{1, \ldots, k\} : x_{ij} \notin [a_j, b_j]$, and by definition of A, for this j we must have $x_{ij} \notin [x_j^{(aj)}, x_j^{(bj)}]$. Now fix any $h \in \mathcal{F}$, and express $\{x : h(x) = +1\} = \times_{j'=1}^{k} [a'_{j'}, b'_{j'}]$. If $h(x_{i'}) = +1$ for all $x_{i'} \in B$, then we must have $a'_{j'} \leq a_{j'}$ and $b'_{j'} \geq b_{j'}$ for every $j' \in \{1, \ldots, k\}$. Furthermore, if $h(x_i) = +1$, then we must have $a'_j \leq x_{ij} \leq b'_j$; but then we must have either $a'_j \leq x_{ij} < x_j^{(aj)}$ or $x_j^{(bj)} < x_{ij} \leq b'_j$. In the former case, since $x_{ij} < x_j^{(aj)}$, we must have $x_j^{(aj)} > 0$, so that there exists a point $x_{i'} \in \mathcal{U}$ with $x_{i'j} = x_j^{(aj)}$ and with $x_{i'j'} \in [a_{j'}, b_{j'}]$ for all $j' \neq j$, and furthermore (by definition of A), $x_{i'} \in A$; but since $[a_{j'}, b_{j'}] \subseteq [a'_{j'}, b'_{j'}]$ we also have $x_{i'j'} \in [a'_{j'}, b'_{j'}]$ for all $j' \neq j$, and since $a'_j < x_j^{(aj)} = x_{i'j} < a_j \leq b_j \leq b'_j$, we also have $x_{i'j} \in [a'_j, b'_j]$. Altogether, we must have $h(x_{i'}) = +1$, which proves there exists at least one point in $A \cup B$ classified differently by h and f^*. The case that $x_j^{(bj)} < x_{ij} \leq b'_j$ is symmetric to this one, so that by the same reasoning, this h must disagree with f^* on the classification of some point in $A \cup B$. Therefore, every $h \in \mathcal{F}$ with $h(x) = f^*(x)$ for all $x \in A \cup B$ has $h(x_i) = -1$ for all $x_i \in \mathcal{U}^- \setminus (A \cup B)$. Combined with the above proof that every such h also has $h(x_i) = +1$ for every $x_i \in \mathcal{U}^+$, we have that every such h has $h(x) = f^*(x)$ for every $x \in \mathcal{U}$. ■

One implication of Theorem 14, combined with Theorem 4, is that

$$\theta(r_0) \leq 128 \frac{k}{\lambda} \ln(160k)$$

for all $r_0 \geq 0$, for P and $\mathcal{F}$ as in Theorem 14. This has implications, both for the label complexity of CAL (via Lemma 30), and also for the label complexity of noise-robust disagreement-based methods (see Section 6 below). More directly, combining Theorem 14 with Theorem 10 yields the following label complexity bound for CAL, which improves over the best previously published bound on the label complexity of CAL for this problem (due to El-Yaniv and Wiener, 2012), reducing the dependence on k from $\Theta(k^3 \log^2(k))$ to $\Theta(k \log^2(k))$.

Corollary 15 *There exists a finite universal constant $c > 0$ such that, for $k \in \mathbb{N}$ and $\lambda \in (0,1)$, for any P with marginal distribution over $\mathcal{X}$ that is a product distribution with marginals having continuous CDFs, and for $\mathcal{F}$ the space of axis-aligned rectangles h on $\mathbb{R}^k$ with $P((x,y) : h(x) = 1) \geq \lambda$, $\forall \varepsilon, \delta \in (0,1/2)$,*

$$\Lambda(\varepsilon,\delta) \leq c \frac{k}{\lambda} \log\left(\frac{k}{\delta}\log\left(\frac{1}{\varepsilon}\right)\right) \log\left(\frac{k}{\varepsilon}\log\left(\frac{1}{\delta}\right)\right) \log\left(\frac{\lambda \log(1/\varepsilon)}{\varepsilon \log(k)} \vee e\right).$$

Proof The result follows by plugging the bound from Theorem 14 into Theorem 10, taking $\delta_m = \delta/(2\log_2(2M(\varepsilon,\delta/2)))$, bounding $M(\varepsilon,\delta/2) \leq \frac{8k}{\varepsilon}\log(\frac{8e}{\varepsilon}) + \frac{8}{\varepsilon}\log(\frac{24}{\delta})$ (Vapnik, 1982; Anthony and Bartlett, 1999), and simplifying the resulting expression. ■

This result is particularly interesting in light of the following lower bound on the label complexities achievable by *any* active learning algorithm.

Theorem 16 *For $k \in \mathbb{N} \setminus \{1\}$ and $\lambda \in (0,1/4]$, letting P_X denote the uniform probability distribution over $(0,1)^k$, for $\mathcal{F}$ the space of axis-aligned rectangles h on $\mathbb{R}^k$ with $P_X(x : h(x) = 1) \geq \lambda$, for any active learning algorithm $\mathcal{A}$, $\forall \delta \in (0,1/2]$, $\forall \varepsilon \in (0,1/(8k))$, there exists a function $f^* \in \mathcal{F}$ such that, if P is the realizable-case distribution having marginal P_X over $\mathcal{X}$ and having target function f^*, if $\mathcal{A}$ is allowed fewer than*

$$\max\left\{k\log\left(\frac{1}{4k\varepsilon}\right), (1-\delta)\left\lfloor\frac{1}{\varepsilon \vee \lambda}\right\rfloor\right\} - 1$$

label requests, then with probability greater than δ, the returned classifier $\hat{h}$ has $\mathrm{er}(\hat{h}) > \varepsilon$.

Proof For any $\varepsilon > 0$, let $\mathcal{M}(\varepsilon)$ denote the maximum number M of classifiers $h_1, \ldots, h_M \in \mathcal{F}$ such that, $\forall i, j \leq M$ with $i \neq j$, $P_X(x : h_i(x) \neq h_j(x)) \geq 2\varepsilon$. Kulkarni, Mitter, and Tsitsiklis (1993) prove that, for any learning algorithm based on binary-valued queries, with a budget smaller than $\log_2((1-\delta)\mathcal{M}(2\varepsilon))$ queries, there exists a target function $f^* \in \mathcal{F}$ such that the classifier $\hat{h}$ produced by the algorithm (when P has marginal P_X over $\mathcal{X}$ and has target function f^*) will have $\mathrm{er}(\hat{h}) > \varepsilon$ with probability greater than δ. In particular, since active learning queries are binary-valued in the binary classification setting, this lower bound applies to active learning algorithms as a special case.

Thus, for the first term in the lower bound, we focus on establishing a lower bound on $\mathcal{M}(2\varepsilon)$ for this problem. First note that $(1-1/k)^k \geq 1/4$, so that $\lambda \leq (1-1/k)^k$. Furthermore, $(1/k)(1-1/k)^{k-1} > 1/(4k)$, so that $\varepsilon < (1/k)(1-1/k)^{k-1}$. Now let

$$\mathcal{F}_{2\varepsilon} = \left\{(x_1,\ldots,x_k) \mapsto 2\prod_{j=1}^{k} \mathbb{1}_{[a_j,b_j]}(x_j) - 1 : \forall j \leq k, b_j = a_j + 1 - 1/k, \right.$$
$$\left. a_j \in \left\{0, \frac{\varepsilon}{(1-1/k)^{k-1}}, \ldots, \left\lfloor\frac{(1-1/k)^{k-1}}{\varepsilon k}\right\rfloor \frac{\varepsilon}{(1-1/k)^{k-1}}\right\}\right\}.$$

Note that $|\mathcal{F}_{2\varepsilon}| = \left(1 + \left\lfloor \frac{(1-1/k)^{k-1}}{\varepsilon k} \right\rfloor\right)^k$. Furthermore, since every $a_j \in [0, 1/k]$ in the specification of $\mathcal{F}_{2\varepsilon}$, we have $b_j = a_j + 1 - 1/k \in [0,1]$, which implies $P_X((x_1,\ldots,x_k) : \prod_{j=1}^k \mathbb{1}_{[a_j,b_j]}(x_j) = 1) = (1-1/k)^k \geq \lambda$. Therefore, $\mathcal{F}_{2\varepsilon} \subseteq \mathcal{F}$. Finally, for each $\{(a_j,b_j)\}_{j=1}^k$ and $\{(a'_j,b'_j)\}_{j=1}^k$ specifying distinct classifiers in $\mathcal{F}_{2\varepsilon}$, at least one j has $|a_j - a'_j| \geq \frac{\varepsilon}{(1-1/k)^{k-1}}$. Since all of the elements $h \in \mathcal{F}_{2\varepsilon}$ have $P_X(x : h(x) = +1) = (1-1/k)^k$, we can note that

$$\begin{aligned}
&P_X\left((x_1,\ldots,x_k) : \prod_{i=1}^k \mathbb{1}_{[a_i,b_i]}(x_i) \neq \prod_{i=1}^k \mathbb{1}_{[a'_i,b'_i]}(x_i)\right) \\
&= 2(1-1/k)^k - 2P_X\left((\times_{i=1}^k [a_i,b_i]) \cap (\times_{i=1}^k [a'_i,b'_i])\right) \\
&= 2(1-1/k)^k - 2P_X\left(\times_{i=1}^k [\max\{a_i,a'_i\}, \min\{b_i,b'_i\}]\right) \\
&= 2(1-1/k)^k - 2\prod_{i=1}^k (\min\{b_i,b'_i\} - \max\{a_i,a'_i\}).
\end{aligned}$$

Thus, since

$$\begin{aligned}
&\prod_{i=1}^k (\min\{b_i,b'_i\} - \max\{a_i,a'_i\}) \\
&\leq (\min\{b_j,b'_j\} - \max\{a_j,a'_j\}) \prod_{i\neq j} (b_i - a_i) = (1-1/k)^{k-1}(\min\{b_j,b'_j\} - \max\{a_j,a'_j\}) \\
&= (1-1/k)^{k-1}(\min\{a_j,a'_j\} - \max\{a_j,a'_j\} + (1-1/k)) = (1-1/k)^{k-1}(1-1/k - |a_j - a'_j|) \\
&\leq (1-1/k)^{k-1}(1-1/k - \frac{\varepsilon}{(1-1/k)^{k-1}}) = (1-1/k)^k - \varepsilon,
\end{aligned}$$

we have

$$P_X((x_1,\ldots,x_k) : \prod_{i=1}^k \mathbb{1}_{[a_i,b_i]}(x_i) \neq \prod_{i=1}^k \mathbb{1}_{[a'_i,b'_i]}(x_i)) \geq 2(1-1/k)^k - 2((1-1/k)^k - \varepsilon) = 2\varepsilon.$$

Thus, $\mathcal{M}(2\varepsilon) \geq \left(1 + \left\lfloor \frac{(1-1/k)^{k-1}}{\varepsilon k} \right\rfloor\right)^k$. Finally, note that for $\delta \in (0, 1/2]$, this implies

$$\log_2((1-\delta)\mathcal{M}(2\varepsilon)) \geq k \log_2\left(\frac{(1-1/k)^{k-1}}{\varepsilon k}\right) - 1 \geq k \log_2\left(\frac{1}{4k\varepsilon}\right) - 1.$$

Together with the aforementioned lower bound of Kulkarni, Mitter, and Tsitsiklis (1993), this establishes the first term in the lower bound.

To prove the second term, we use of a technique of Hanneke (2007b). Specifically, fix any finite set $H \subseteq \mathcal{F}$ with $\min_{h,g\in H} P_X(x : h(x) \neq g(x)) \geq 2\varepsilon$, let

$$\mathrm{XPTD}(f, H, \mathcal{U}, \delta) = \min\{t \in \mathbb{N} : \exists R \subseteq \mathcal{U} : |R| \leq t, |\{h \in H : \forall x \in R, h(x) = f(x)\}| \leq \delta|H| + 1\} \cup \{\infty\},$$

for any classifier f and $\mathcal{U} \in \bigcup_m \mathcal{X}^m$, and let $\mathrm{XPTD}(H, P_X, \delta)$ denote the smallest $t \in \mathbb{N}$ such that every classifier f has $\lim_{m\to\infty} \mathbb{P}_{\mathcal{U}\sim P_X^m}(\mathrm{XPTD}(f, H, \mathcal{U}, \delta) > t) = 0$. Then Hanneke (2007b) proves that there exists a choice of target function $f^* \in \mathcal{F}$ for the distribution P such that, if $\mathcal{A}$ is allowed fewer than $\mathrm{XPTD}(H, P_X, \delta)$ label requests, then with probability greater than δ, the returned

classifier $\hat{h}$ has $\mathrm{er}(\hat{h}) > \varepsilon$. For the particular problem studied here, let H be the set of classifiers $h_i(x) = 2\mathbb{1}_{[(i-1)(\varepsilon\vee\lambda),i(\varepsilon\vee\lambda)]\times[0,1]^{k-1}}(x) - 1$, for $i \in \left\{1,\ldots,\left\lfloor\frac{1}{\varepsilon\vee\lambda}\right\rfloor\right\}$. Note that each $h_i \in H$ has $P_X(x : h_i(x) = +1) = P_X((x_1,\ldots,x_k) : x_1 \in [(i-1)(\varepsilon\vee\lambda), i(\varepsilon\vee\lambda)]) = \varepsilon\vee\lambda \geq \lambda$, so that $H \subseteq \mathcal{F}$. Furthermore, for any $h_i, h_j \in H$ with $i \neq j$, $P_X(x : h_i(x) \neq h_j(x)) \geq P_X((x_1,\ldots,x_k) : x_1 \in ((i-1)(\varepsilon\vee\lambda), i(\varepsilon\vee\lambda)) \cup ((j-1)(\varepsilon\vee\lambda), j(\varepsilon\vee\lambda))) = 2(\varepsilon\vee\lambda) \geq 2\varepsilon$. Also, let $R \subseteq (0,1)^k$ be any finite set with no points $(x_1,\ldots,x_k) \in R$ such that $x_1 \in \left\{i(\varepsilon\vee\lambda) : i \in \left\{1,\ldots,\left\lfloor\frac{1}{\varepsilon\vee\lambda}\right\rfloor - 1\right\}\right\}$; note that every $x \in R$ has exactly one $h_i \in H$ with $h_i(x) = +1$. Thus, for the classifier f with $f(x) = -1$ for all $x \in \mathcal{X}$, $|\{h \in H : \forall x \in R, h(x) = f(x)\}| \geq |H| - |R|$. Thus, for any set $\mathcal{U} \subseteq (0,1)^k$ with no points $(x_1,\ldots,x_k) \in \mathcal{U}$ having $x_1 \in \left\{i(\varepsilon\vee\lambda) : i \in \left\{1,\ldots,\left\lfloor\frac{1}{\varepsilon\vee\lambda}\right\rfloor - 1\right\}\right\}$, we have $\mathrm{XPTD}(f,H,\mathcal{U},\delta) \geq (1-\delta)|H| - 1$. Since, for all $m \in \mathbb{N}$, the probability that $\mathcal{U} \sim P_X^m$ contains a point $(x_1,\ldots,x_k)$ with $x_1 \in \left\{i(\varepsilon\vee\lambda) : i \in \left\{1,\ldots,\left\lfloor\frac{1}{\varepsilon\vee\lambda}\right\rfloor - 1\right\}\right\}$ is zero, we have that $\mathbb{P}_{\mathcal{U}\sim P_X^m}(\mathrm{XPTD}(f,H,\mathcal{U},\delta) \geq (1-\delta)|H| - 1) = 1$. This implies $\mathrm{XPTD}(H,P_X,\delta) \geq (1-\delta)|H| - 1 = (1-\delta)\left\lfloor\frac{1}{\varepsilon\vee\lambda}\right\rfloor - 1$. Combining this with the lower bound of Hanneke (2007b) implies the result. ■

Together, Corollary 15 and Theorem 16 imply that, for $\lambda \in (0, 1/4]$ bounded away from 0, the label complexity of CAL is within logarithmic factors of the minimax optimal label complexity.

6. New Label Complexity Bounds for Agnostic Active Learning

In this section we present new bounds on the label complexity of noise-robust active learning algorithms, expressed in terms of $\mathcal{B}_{\hat{n}}(m,\delta)$. These bounds yield new exponential label complexity speedup results for agnostic active learning (for the low accuracy regime) of linear classifiers under a fixed mixture of Gaussians. Analogous results also hold for the problem of learning axis-aligned rectangles under a product density.

Specifically, in the *agnostic* setting studied in this section, we no longer assume $\exists f^* \in \mathcal{F}$ with $\mathbb{P}(Y = f^*(x)|X) = 1$ for $(X,Y) \sim P$, but rather allow that P is *any* probability measure over $\mathcal{X}\times\mathcal{Y}$. In this setting, we let $f^* : \mathcal{X} \to \mathcal{Y}$ denote a classifier such that $\mathrm{er}(f^*) = \inf_{h\in\mathcal{F}} \mathrm{er}(h)$ and $\inf_{h\in\mathcal{F}} P((x,y) : h(x) \neq f^*(x)) = 0$, which is guaranteed to exist by topological considerations (see Hanneke, 2012, Section 6.1);[8] for simplicity, when $\exists f \in \mathcal{F}$ with $\mathrm{er}(f) = \inf_{h\in\mathcal{F}} \mathrm{er}(h)$, we take f^* to be an element of $\mathcal{F}$. We call f^* the *infimal* hypothesis (of $\mathcal{F}$, w.r.t. P) and note that $\mathrm{er}(f^*)$ is sometimes called the *noise rate of* $\mathcal{F}$ (e.g., Balcan, Beygelzimer, and Langford, 2006). The introduction of the infimal hypothesis f^* allows for natural generalizations of some of the key definitions of Section 2 that facilitate analysis in the agnostic setting.

Definition 17 (Agnostic Version Space) *Let f^* be the infimal hypothesis of $\mathcal{F}$ w.r.t. P. The* agnostic version space *of a sample S is*

$$\mathrm{VS}_{\mathcal{F},S,f^*} \triangleq \{h \in \mathcal{F} : \forall (x,y) \in S, h(x) = f^*(x)\}.$$

Definition 18 (Agnostic Version Space Compression Set Size) *Letting $\hat{\mathcal{C}}_{S,f^*}$ denote a smallest subset of S satisfying $\mathrm{VS}_{\mathcal{F},\hat{\mathcal{C}}_{S,f^*},f^*} = \mathrm{VS}_{\mathcal{F},S,f^*}$, the* agnostic version space compression set size *is*

$$\hat{n}(\mathcal{F},S,f^*) \triangleq |\hat{\mathcal{C}}_{S,f^*}|.$$

8. In the agnostic setting, there are typically many valid choices of the function f^* satisfying these conditions. The results below hold for *any* such choice of f^*.

We also extend the definition of the version space compression set minimal *bound* (see Definition 2) to the agnostic setting, defining

$$\mathcal{B}_{\hat{n}}(m,\delta) \triangleq \min\{b \in \mathbb{N} \cup \{0\} : \mathbb{P}(\hat{n}(\mathcal{F},S,f^*) \leq b) \geq 1-\delta\}.$$

For general P in the agnostic setting, define the disagreement coefficient as before, except now with respect to the infimal hypothesis:

$$\theta(r_0) \triangleq \sup_{r>r_0} \frac{\Delta \mathrm{B}(f^*,r)}{r} \vee 1.$$

One can easily verify that these definitions are equal to those given above in the special case that P satisfies the realizable-case assumptions ($f^* \in \mathcal{F}$ and $\mathbb{P}(Y = f^*(X)|X) = 1$ for $(X,Y) \sim P$).

We begin with the following extension of Theorem 4.

Lemma 19 *For general (agnostic) P, for any $r_0 \in (0,1)$,*

$$\theta(r_0) \leq \max\left\{ \max_{r\in(r_0,1)} 16\mathcal{B}_{\hat{n}}\left(\left\lceil \frac{1}{r} \right\rceil, \frac{1}{20}\right), 512 \right\}.$$

Proof First note that $\theta(r_0)$ and $\mathcal{B}_{\hat{n}}\left(\left\lceil \frac{1}{r} \right\rceil, \frac{1}{20}\right)$ depend on P only via f^* and the marginal $P(\cdot \times \mathcal{Y})$ of P over $\mathcal{X}$ (in both the realizable case and agnostic case). Define a distribution P' with marginal $P'(\cdot \times \mathcal{Y}) = P(\cdot \times \mathcal{Y})$ over $\mathcal{X}$, and with $\mathbb{P}(Y = f^*(x)|X = x) = 1$ for all $x \in \mathcal{X}$, where $(X,Y) \sim P'$. In particular, in the special case that $f^* \in \mathcal{F}$ in the agnostic case, we have that P' is a distribution in the realizable case, with identical values of $\theta(r_0)$ and $\mathcal{B}_{\hat{n}}\left(\left\lceil \frac{1}{r} \right\rceil, \frac{1}{20}\right)$ as P, so that Theorem 4 (applied to P') implies the result. On the other hand, when P is a distribution with $f^* \notin \mathcal{F}$, let $\theta'(r_0)$ denote the disagreement coefficient of $\mathcal{F} \cup \{f^*\}$ with respect to P' (or equivalently P), and for $m \in \mathbb{N}$, let $\mathcal{B}'_{\hat{n}}(m,1/20) \triangleq \min\{b \in \mathbb{N} \cup \{0\} : \mathbb{P}(\hat{n}(\mathcal{F} \cup \{f^*\}, S_m, f^*) \leq b) \geq 19/20\}$. In particular, since $\mathcal{F} \subseteq \mathcal{F} \cup \{f^*\}$, we have $\theta(r_0) \leq \theta'(r_0)$, and since P' is a realizable-case distribution with respect to the hypothesis class $\mathcal{F} \cup \{f^*\}$, Theorem 4 (applied to P' and $\mathcal{F} \cup \{f^*\}$) implies

$$\theta'(r_0) \leq \max\left\{ \max_{r\in(r_0,1)} 16\mathcal{B}'_{\hat{n}}\left(\left\lceil \frac{1}{r} \right\rceil, \frac{1}{20}\right), 512 \right\}.$$

Finally, note that for any $m \in \mathbb{N}$ and sets $C,S \in (\mathcal{X} \times \mathcal{Y})^m$, $\mathrm{VS}_{\mathcal{F}\cup\{f^*\},C,f^*} = \mathrm{VS}_{\mathcal{F},C,f^*} \cup \{f^*\}$ and $\mathrm{VS}_{\mathcal{F}\cup\{f^*\},S,f^*} = \mathrm{VS}_{\mathcal{F},S,f^*} \cup \{f^*\}$, so that $\mathrm{VS}_{\mathcal{F}\cup\{f^*\},C,f^*} = \mathrm{VS}_{\mathcal{F}\cup\{f^*\},S,f^*}$ if and only if $\mathrm{VS}_{\mathcal{F},C,f^*} = \mathrm{VS}_{\mathcal{F},S,f^*}$. Thus, $\hat{n}(\mathcal{F} \cup \{f^*\}, S_m, f^*) = \hat{n}(\mathcal{F}, S_m, f^*)$, so that $\mathcal{B}'_{\hat{n}}\left(\left\lceil \frac{1}{r} \right\rceil, \frac{1}{20}\right) = \mathcal{B}_{\hat{n}}\left(\left\lceil \frac{1}{r} \right\rceil, \frac{1}{20}\right)$, which implies the result. ■

6.1 Label complexity bound for agnostic active learning

A^2 (*Agnostic Active*) was the first general-purpose agnostic active learning algorithm with proven improvement in error guarantees compared to passive learning. The original work of Balcan, Beygelzimer, and Langford (2006), which first introduced this algorithm, also provided specialized proofs that the algorithm achieves an exponential label complexity speedup (for the low accuracy regime) compared to passive learning for a few simple cases, including: threshold functions, and

homogeneous linear separators under a uniform distribution over the sphere. Additionally, Hanneke (2007a) provided a general bound on the label complexity of A^2, expressed in terms of the disagreement coefficient, so that any bound on the disagreement coefficient translates into a bound on the label complexity of agnostic active learning with A^2. Inspired by the A^2 algorithm, other noise-robust active learning algorithms have since been proposed, with improved label complexity bounds compared to those proven by Hanneke (2007a) for A^2, while still expressed in terms of the disagreement coefficient (see e.g., Dasgupta, Hsu, and Monteleoni, 2007; Hanneke, 2014). As an example of such results, the following result was proven by Dasgupta, Hsu, and Monteleoni (2007).

Theorem 20 (Dasgupta, Hsu, and Monteleoni, 2007) *There exists a finite universal constant $c > 0$ such that, for any $\varepsilon, \delta \in (0, 1/2)$, using hypothesis class $\mathcal{F}$, and given the input δ and a budget n on the number of label requests, the active learning algorithm of Dasgupta, Hsu, and Monteleoni (2007) requests at most n labels,*[9] *and if*

$$n \geq c\theta(\mathrm{er}(f^*)+\varepsilon)\left(\frac{\mathrm{er}(f^*)^2}{\varepsilon^2}+1\right)\left(d\log\left(\frac{1}{\varepsilon}\right)+\log\left(\frac{1}{\delta}\right)\right)\log\left(\frac{1}{\varepsilon}\right),$$

then with probability at least $1-\delta$, the classifier $\hat{f} \in \mathcal{F}$ it produces satisfies

$$\mathrm{er}(\hat{f}) \leq \mathrm{er}(f^*)+\varepsilon.$$

Combined with the results above, this implies the following theorem.

Theorem 21 *There exists a finite universal constant $c > 0$ such that, for any $\varepsilon, \delta \in (0, 1/2)$, using hypothesis class $\mathcal{F}$, and given the input δ and a budget n on the number of label requests, the active learning algorithm of Dasgupta, Hsu, and Monteleoni (2007) requests at most n labels, and if*

$$n \geq c\left(\max_{r>\mathrm{er}(f^*)+\varepsilon}\mathcal{B}_{\hat{n}}\left(\left\lceil\frac{1}{r}\right\rceil,\frac{1}{20}\right)+1\right)\left(\frac{\mathrm{er}(f^*)^2}{\varepsilon^2}+1\right)\left(d\log\left(\frac{1}{\varepsilon}\right)+\log\left(\frac{1}{\delta}\right)\right)\log\left(\frac{1}{\varepsilon}\right),$$

then with probability at least $1-\delta$, the classifier $\hat{f} \in \mathcal{F}$ it produces satisfies

$$\mathrm{er}(\hat{f}) \leq \mathrm{er}(f^*)+\varepsilon.$$

Proof By Lemma 19,

$$\begin{aligned}\theta(\mathrm{er}(f^*)+\varepsilon) &\leq \max\left\{\max_{r\in(\mathrm{er}(f^*)+\varepsilon,1)} 16\mathcal{B}_{\hat{n}}\left(\left\lceil\frac{1}{r}\right\rceil,\frac{1}{20}\right), 512\right\}\\ &\leq 512\left(\max_{r>\mathrm{er}(f^*)+\varepsilon}\mathcal{B}_{\hat{n}}\left(\left\lceil\frac{1}{r}\right\rceil,\frac{1}{20}\right)+1\right).\end{aligned}$$

Plugging this into Theorem 20 yields the result. ■

9. This result applies to a slightly modified variant of the algorithm of Dasgupta, Hsu, and Monteleoni (2007), studied by Hanneke (2011), which terminates after a given number of label requests, rather than after a given number of unlabeled samples. The same is true of Theorem 21 and Corollary 22.

Interestingly, from the perspective of bounding the label complexity of agnostic active learning in general, the result in Theorem 21 sometimes improves over a related bound proven by Hanneke (2007b) (for a different algorithm). Specifically, compared to the result of Hanneke (2007b), this result maintains an interesting dependence on f^*, whereas the bound of Hanneke (2007b) effectively replaces the factor $\mathcal{B}_{\hat{\eta}}(\lceil 1/r \rceil, 1/20)$ with the maximum of this quantity over the choice of f^*.[10] Also, while the result of Hanneke (2007b) is proven for an algorithm that requires explicit access to a value $\eta \approx \mathrm{er}(f^*)$ to obtain the stated label complexity, the label complexity in Theorem 21 is achieved by the algorithm of Dasgupta, Hsu, and Monteleoni (2007), which requires no such extra parameters.

As an application of Theorem 21, we have the following corollary.

Corollary 22 *For $t, k \in \mathbb{N}$ and $c \in (0,\infty)$, there exists a finite constant $c_{k,t,c} > 0$ such that, for $\mathcal{F}$ the class of linear separators on $\mathbb{R}^k$, and for P with marginal distribution over $\mathcal{X}$ that is a mixture of t multivariate normal distributions with diagonal covariance matrices of full rank, for any $\varepsilon, \delta \in (0, 1/2)$ with $\varepsilon \geq \frac{\mathrm{er}(f^*)}{c}$, using hypothesis class $\mathcal{F}$, and given the input δ and a budget n on the number of label requests, the active learning algorithm of Dasgupta, Hsu, and Monteleoni (2007) requests at most n labels, and if*

$$n \geq c_{k,t,c} \left(\log\left(\frac{1}{\varepsilon}\right) \right)^{k+1} \log\left(\frac{1}{\delta}\right),$$

then with probability at least $1-\delta$, the classifier $\hat{f} \in \mathcal{F}$ it produces satisfies $\mathrm{er}(\hat{f}) \leq \mathrm{er}(f^*) + \varepsilon$.

Proof Let $\mathcal{F}$ and P be as described above. First, we argue that $f^* \in \mathcal{F}$. Fix any classifier f with $\inf_{h \in \mathcal{F}} P((x,y) : h(x) \neq f(x)) = 0$. There must exist a sequence $\{(b^{(t)}, w_1^{(t)}, \ldots, w_k^{(t)})\}_{k=1}^{\infty}$ in $\mathbb{R}^{k+1}$ with $\sum_{i=1}^k (w_i^{(t)})^2 = 1$ for all t, s.t. $P\left((x_1,\ldots,x_k,y) : \mathrm{sign}\left(b^{(t)} + \sum_{i=1}^k x_i w_i^{(t)}\right) \neq f(x_1,\ldots,x_k)\right) \to 0$. If $\limsup_{t\to\infty} b^{(t)} = \infty$, then $\exists t_j \to \infty$ with $b^{(t_j)} \to \infty$, and since every $(x_1,\ldots,x_k) \in \mathbb{R}^k$ has $\sum_{i=1}^k x_i w_i^{(t)} \geq -\|x\|$, we have that $b^{(t_j)} + \sum_{i=1}^k x_i w_i^{(t_j)} \to \infty$, which implies $\mathrm{sign}\left(b^{(t_j)} + \sum_{i=1}^k x_i w_i^{(t_j)}\right) \to 1$ for all $(x_1,\ldots,x_k) \in \mathbb{R}^k$. Similarly, if $\liminf_{t\to\infty} b^{(t)} = -\infty$, then $\exists t_j \to \infty$ with $\mathrm{sign}\left(b^{(t_j)} + \sum_{i=1}^k x_i w_i^{(t_j)}\right) \to -1$ for all $(x_1,\ldots,x_k) \in \mathbb{R}^k$. Otherwise, if $\limsup_{t\to\infty} b^{(t)} < \infty$ and $\liminf_{t\to\infty} b^{(t)} > -\infty$, then the sequence $\{(b^{(t)}, w_1^{(t)}, \ldots, w_k^{(t)})\}_{t=1}^{\infty}$ is *bounded* in $\mathbb{R}^{k+1}$. Therefore, the Bolzano-Weierstrass Theorem implies it contains a convergent subsequence: that is, $\exists t_j \to \infty$ s.t. $(b^{(t_j)}, w_1^{(t_j)}, \ldots, w_k^{(t_j)})$ converges. Furthermore, since $\{w \in \mathbb{R}^k : \|w\| = 1\}$ is closed, and $\{b^{(t)} : t \in \mathbb{N}\} \subseteq [\inf_t b^{(t)}, \sup_t b^{(t)}]$, which is a closed subset of $\mathbb{R}$, $\exists (b, w_1, \ldots, w_k) \in \mathbb{R}^{k+1}$ with $\sum_{i=1}^k w_i^2 = 1$ such that $(b^{(t_j)}, w_1^{(t_j)}, \ldots, w_k^{(t_j)}) \to (b, w_1, \ldots, w_k)$. Continuity of linear functions implies, $\forall (x_1,\ldots,x_k) \in \mathbb{R}^k$, $b^{(t_j)} + \sum_{i=1}^k x_i w_i^{(t_j)} \to b + \sum_{i=1}^k x_i w_i$. Therefore, every $(x_1,\ldots,x_k) \in \mathbb{R}^k$ with $b + \sum_{i=1}^k x_i w_i > 0$ has $\mathrm{sign}\left(b^{(t_j)} + \sum_{i=1}^k x_i w_i^{(t_j)}\right) \to 1$, and every $(x_1,\ldots,x_k) \in \mathbb{R}^k$ with $b + \sum_{i=1}^k x_i w_i < 0$ has $\mathrm{sign}\left(b^{(t_j)} + \sum_{i=1}^k x_i w_i^{(t_j)}\right) \to -1$. Since $P\left((x_1,\ldots,x_k,y) : b + \sum_{i=1}^k x_i w_i = 0\right) = 0$, this implies $(x_1,\ldots,x_k) \mapsto \mathrm{sign}\left(b^{(t_j)} + \sum_{i=1}^k x_i w_i^{(t_j)}\right)$ converges to $(x_1,\ldots,x_k) \mapsto \mathrm{sign}\left(b + \sum_{i=1}^k x_i w_i\right)$ almost surely $[P]$.

10. There are a few other differences, which are usually minor. For instance, the bound of Hanneke (2007b) uses $r \approx \mathrm{er}(f^*) + \varepsilon$ rather than maximizing over $r > \mathrm{er}(f^*) + \varepsilon$. That result additionally replaces "1/20" with a value $\delta' \approx \delta/n$.

Thus, in each case, $\exists t_j \to \infty$ and $h \in \mathcal{F}$ s.t. $(x_1,\ldots,x_k) \mapsto \text{sign}\left(b^{(t_j)} + \sum_{i=1}^k x_i w_i^{(t_j)}\right)$ converges to h a.s. $[P]$. Since convergence almost surely implies convergence in probability, we have $P\left((x_1,\ldots,x_k,y) : \text{sign}\left(b^{(t_j)} + \sum_{i=1}^k x_i w_i^{(t_j)}\right) \neq h(x_1,\ldots,x_k)\right) \to 0$. Furthermore, by assumption, $P\left((x_1,\ldots,x_k,y) : \text{sign}\left(b^{(t_j)} + \sum_{i=1}^k x_i w_i^{(t_j)}\right) \neq f(x_1,\ldots,x_k)\right) \to 0$ as well. Thus, a union bound implies $P((x,y) : h(x) \neq f(x)) = 0$. In particular, we have that for any f with $\inf_{g\in\mathcal{F}} P((x,y) : g(x) \neq f(x)) = 0$ and $\text{er}(f) = \inf_{g\in\mathcal{F}} \text{er}(g)$, $\exists h \in \mathcal{F}$ with $P((x,y) : f(x) \neq h(x)) = 0$, and hence $\text{er}(h) = \inf_{g\in\mathcal{F}} \text{er}(g)$. Thus, we may assume $f^* \in \mathcal{F}$ in this setting.

Therefore, in this scenario, Theorem 12 implies

$$\max_{r>\text{er}(f^*)+\varepsilon} \mathcal{B}_{\hat{n}}\left(\left\lceil \frac{1}{r} \right\rceil, \frac{1}{20}\right) + 1 \leq c_{k,t}^{(1)} \left(\log\left(\frac{2}{\text{er}(f^*)+\varepsilon}\right)\right)^{k-1},$$

for an appropriate (k,t)-dependent constant $c_{k,t}^{(1)} \in (0,\infty)$. Plugging this into Theorem 21, and recalling that the VC dimension of the class of linear classifiers in $\mathbb{R}^k$ is $k+1$ (see e.g., Anthony and Bartlett, 1999), we get a bound on the number of label requests of

$$c_{k,t}^{(2)} \left(\log\left(\frac{2}{\text{er}(f^*)+\varepsilon}\right)\right)^{k-1} \left(\frac{\text{er}(f^*)^2}{\varepsilon^2}+1\right)\left(k\log\left(\frac{1}{\varepsilon}\right)+\log\left(\frac{1}{\delta}\right)\right)\log\left(\frac{1}{\varepsilon}\right)$$
$$\leq c_{k,t}^{(3)} \left(\log\left(\frac{1}{\varepsilon}\right)\right)^{k+1} \left(\frac{\text{er}(f^*)^2}{\varepsilon^2}+1\right)\left(k+\log\left(\frac{1}{\delta}\right)\right),$$

for appropriate (k,t)-dependent constants $c_{k,t}^{(2)}, c_{k,t}^{(3)} \in (0,\infty)$. Since (by assumption) $\varepsilon \geq \frac{\text{er}(f^*)}{c}$, this is at most

$$c_{k,t,c}^{(4)} \left(\log\left(\frac{1}{\varepsilon}\right)\right)^{k+1} \left(k+\log\left(\frac{1}{\delta}\right)\right) \leq c_{k,t,c}^{(5)} \left(\log\left(\frac{1}{\varepsilon}\right)\right)^{k+1} \log\left(\frac{1}{\delta}\right),$$

for appropriate (k,t,c)-dependent constants $c_{k,t,c}^{(4)}, c_{k,t,c}^{(5)} \in (0,\infty)$. Thus, taking $c_{k,t,c} = c_{k,t,c}^{(5)}$ establishes the result. ■

An analogous result can be shown for the problem of learning axis-aligned rectangles via Theorem 14.

6.2 Label complexity bound under Mammen-Tsybakov noise

Since the original work on agnostic active learning discussed above, there have been several other analyses, expressing the noise conditions in terms of quantities other than the noise rate $\text{er}(f^*)$. Specifically, the following condition of Mammen and Tsybakov (1999) has been studied for several algorithms (see e.g., Balcan, Broder, and Zhang, 2007; Hanneke, 2011; Koltchinskii, 2010; Hanneke, 2012; Hanneke and Yang, 2012; Hanneke, 2014; Beygelzimer, Hsu, Langford, and Zhang, 2010; Hsu, 2010).

Condition 23 (Mammen and Tsybakov, 1999) *For some $a \in [1,\infty)$ and $\alpha \in [0,1]$, for every $f \in \mathcal{F}$,*

$$\Pr(f(X) \neq f^*(X)) \leq a(\text{er}(f) - \text{er}(f^*))^\alpha.$$

In particular, for a variant of A^2 known as RobustCAL$_\delta$, studied by Hanneke (2012, 2014) and Hanneke and Yang (2012), the following result is known (due to Hanneke and Yang, 2012).

Theorem 24 (Hanneke and Yang, 2012) *There exists a finite universal constant $c > 0$ such that, for any $\varepsilon, \delta \in (0, 1/2)$, for any $n, u \in \mathbb{N}$, given the arguments n and u, the RobustCAL$_\delta$ algorithm requests at most n labels, and if u is sufficiently large, and*

$$n \geq ca^2\theta(a\varepsilon^\alpha)\left(\frac{1}{\varepsilon}\right)^{2-2\alpha}\left(d\log\left(e\theta\left(a\varepsilon^\alpha\right)\right)+\log\left(\frac{\log(1/\varepsilon)}{\delta}\right)\right)\log\left(\frac{1}{\varepsilon}\right),$$

for a and α as in Condition 23, then with probability at least $1-\delta$, the classifier $\hat{f} \in \mathcal{F}$ it returns satisfies $\mathrm{er}(\hat{f}) \leq \mathrm{er}(f^*)+\varepsilon$.

Combined with Theorem 4, this implies the following theorem.

Theorem 25 *There exists a finite universal constant $c > 0$ such that, for any $\varepsilon, \delta \in (0, 1/2)$, for any $n, u \in \mathbb{N}$, given the arguments n and u, the RobustCAL$_\delta$ algorithm requests at most n labels, and if u is sufficiently large, and*

$$n \geq ca^2\left(\max_{r>a\varepsilon^\alpha}\mathcal{B}_{\hat{n}}\left(\left\lceil\frac{1}{r}\right\rceil,\frac{1}{20}\right)+1\right)\left(\frac{1}{\varepsilon}\right)^{2-2\alpha}\left(d\log\left(\frac{1}{\varepsilon}\right)+\log\left(\frac{1}{\delta}\right)\right)\log\left(\frac{1}{\varepsilon}\right),$$

for a and α as in Condition 23, then with probability at least $1-\delta$, the classifier $\hat{f} \in \mathcal{F}$ it returns satisfies $\mathrm{er}(\hat{f}) \leq \mathrm{er}(f^*)+\varepsilon$.

In particular, reasoning as in Corollary 22 above, Theorem 25 implies the following corollary.

Corollary 26 *For $t, k \in \mathbb{N}$ and $a \in [1, \infty)$, there exists a finite constant $c_{k,t,a} > 0$ such that, for $\mathcal{F}$ the class of linear separators on $\mathbb{R}^k$, and for P satisfying Condition 23 with $\alpha = 1$ and the given value of a, and with marginal distribution over $\mathcal{X}$ that is a mixture of t multivariate normal distributions with diagonal covariance matrices of full rank, for any $\varepsilon, \delta \in (0, 1/2)$, for any $n, u \in \mathbb{N}$, given the arguments n and u, the RobustCAL$_\delta$ algorithm requests at most n labels, and if u is sufficiently large, and*

$$n \geq c_{k,t,a}\left(\log\left(\frac{1}{\varepsilon}\right)\right)^{k+1}\log\left(\frac{1}{\delta}\right),$$

then with probability at least $1-\delta$, the classifier $\hat{f} \in \mathcal{F}$ it returns satisfies $\mathrm{er}(\hat{f}) \leq \mathrm{er}(f^*)+\varepsilon$.

Corollary 26 proves an exponential label complexity speedup in the asymptotic dependence on ε compared to passive learning, for which there is a lower bound on the label complexity of $\Omega(1/\varepsilon)$ in the worst case over these distributions (Long, 1995).

Remark 27 *Condition 23 can be satisfied with $\alpha = 1$ if the Bayes optimal classifier is in $\mathcal{F}$ and the source distribution satisfies* Massart noise *(Massart and Nédélec, 2006):*

$$\Pr\left(|P(Y=1|X=x)-1/2| < 1/(2a)\right) = 0.$$

For example, if the data was generated by some unknown linear hypothesis with label noise (probability to flip any label) of up to $(a-1)/2a$, then P satisfies the requirements of Corollary 26.

Acknowledgements

R. El-Yaniv thanks the Intel Collaborative Research Institute for Computational Intelligence (ICRI-CI) and the Israel Science Foundation for their generous support.

Appendix A. Analysis of CAL via the Disagreement Coefficient

The following result was first established by (Giné and Koltchinskii, 2006, page 1213), with slightly different constant factors. The version stated here is directly from Hanneke (2009, Section 2.9), who also presents a simple and direct proof.

Lemma 28 (Giné and Koltchinskii, 2006; Hanneke, 2009) *For any $t \in \mathbb{N}$ and $\delta \in (0,1)$, with probability at least $1-\delta$,*

$$\sup_{h \in \mathrm{VS}_{\mathcal{F},S_t}} \mathrm{er}(h) \leq \frac{24}{t}\left(d\ln\left(880\cdot\theta(d/t)\right)+\ln\left(\frac{12}{\delta}\right)\right).$$

The following result is implicit in a proof of Hanneke (2011); for completeness, we present a formal proof here.

Lemma 29 (Hanneke, 2011) *There exists a finite universal constant $c_0 > 0$ such that, $\forall \delta \in (0,1)$, $\forall m \in \mathbb{N}$ with $m \geq 2$,*

$$\mathcal{B}_N(m,\delta) \leq c_0\theta(d/m)\left(d\ln\left(e\theta(d/m)\right)+\ln\left(\frac{\log_2(m)}{\delta}\right)\right)\log_2(m).$$

Proof The result trivially holds for $m = 2$, taking any $c_0 \geq 2$. Otherwise, suppose $m \geq 3$. Note that, for any $t \in \mathbb{N}$,

$$\frac{24}{t}\left(d\ln\left(880\theta(d/t)\right)+\ln\left(\frac{24\log_2(m)}{\delta}\right)\right) \leq \frac{c_1}{t}\left(d\ln\left(e\theta(d/t)\right)+\ln\left(\frac{2\log_2(m)}{\delta}\right)\right), \quad (5)$$

for some universal constant $c_1 \in [1,\infty)$ (e.g., taking $c_1 = 168$ suffices). Thus, letting r_t denote the expression on the right hand side of (5), Lemma 28 implies that, for any $t \in \mathbb{N}$, with probability at least $1-\delta/(2\log_2(m))$,

$$\sup_{h \in \mathrm{VS}_{\mathcal{F},S_t}} \mathrm{er}(h) \leq r_t.$$

By a union bound, this holds for all $t \in \{2^i : i \in \{1,\ldots,\lceil\log_2(m)\rceil-1\}\}$ with probability at least $1-\delta/2$. In particular, on this event, we have

$$N(m;S_m) \leq 2+\sum_{i=1}^{\lceil\log_2(m)\rceil-1}\sum_{t=2^i+1}^{2^{i+1}} \mathbb{1}_{\mathrm{DIS}(\mathrm{B}(f^*,r_{2^i}))}(x_t).$$

A Chernoff bound implies that, with probability at least $1-\delta/2$, the right hand side is at most

$$\begin{aligned}
&\log_2\left(\frac{8}{\delta}\right)+2e\sum_{i=1}^{\lceil\log_2(m)\rceil-1}2^i\Delta\mathrm{B}(f^*,r_{2^i})\\
&\leq\log_2\left(\frac{8}{\delta}\right)+2e\sum_{i=1}^{\lceil\log_2(m)\rceil-1}2^i\theta\left(r_{2^i}\right)r_{2^i}\\
&\leq\log_2\left(\frac{8}{\delta}\right)+2ec_1\sum_{i=1}^{\lceil\log_2(m)\rceil-1}\theta\left(d2^{-i}\right)\left(d\ln\left(e\theta\left(d2^{-i}\right)\right)+\ln\left(\frac{2\log_2(m)}{\delta}\right)\right)\\
&\leq 4ec_1\theta(d/m)\left(d\ln\left(e\theta(d/m)\right)+\ln\left(\frac{\log_2(m)}{\delta}\right)\right)\log_2(m).
\end{aligned}$$

Letting $c_0=4ec_1$, the result holds by a union bound and minimality of $\mathcal{B}_N(m,\delta)$. ■

The following result is taken from the work of Hanneke (2011, Proof of Theorem 1); see also Hanneke (2014) for a theorem and proof expressed in this exact form.

Lemma 30 (Hanneke, 2011) *There exists a finite universal constant $c_0>0$ such that, $\forall\varepsilon,\delta\in(0,1/2]$,*

$$\Lambda(\varepsilon,\delta)\leq c_0\theta(\varepsilon)\left(d\ln(e\theta(\varepsilon))+\ln\left(\frac{\log_2(1/\varepsilon)}{\delta}\right)\right)\log_2\left(\frac{1}{\varepsilon}\right).$$

The next result is taken from the work of El-Yaniv and Wiener (2012, Corollary 39).

Lemma 31 (El-Yaniv and Wiener, 2012) *For any $r_0\in(0,1)$,*

$$\theta(r_0)\leq\max\left\{\sup_{r\in(r_0,1/2)}\frac{7\cdot\mathcal{B}_\Delta(\lfloor 1/r\rfloor,1/9)}{r},2\right\}.$$

Appendix B. Separation from the Previous Analyses

There are simple examples showing that sometimes $\mathcal{B}_{\hat{n}}(m,\delta)\approx\theta(1/m)$, so that the upper bound $\Lambda(\varepsilon,\delta)\leq c_0 d\theta(\varepsilon)\mathrm{polylog}\left(\frac{1}{\varepsilon\delta}\right)$ in Lemma 30 is off by a factor of d compared to Theorem 10 in those cases (aside from logarithmic factors). For instance, consider the class of unions of k intervals, where $k\in\mathbb{N}$, $\mathcal{X}=[0,1]$, and $\mathcal{F}=\{x\mapsto 2\mathbb{1}_{\bigcup_{i=1}^k[z_{2i-1},z_{2i}]}(x)-1:0<z_1<\cdots<z_{2k}<1\}$. Suppose the data distribution P has a uniform marginal distribution over $\mathcal{X}$, and has $f^*=2\mathbb{1}_{\bigcup_{i=1}^k[z^*_{2i-1},z^*_{2i}]}-1$, where $z^*_i=\frac{i}{2k+1}$ for $i\in\{1,\ldots,2k\}$. In this case, for $r_0\geq 0$, $\theta(r_0)$ is within a factor of 2 of $\min\left\{\frac{1}{r_0},4k\right\}$ (see e.g., Balcan, Hanneke, and Vaughan, 2010; Hanneke, 2012). However, for any $m\in\mathbb{N}$ with $m\geq(2k+1)\ln\left(\frac{2k+1}{\delta}\right)$, with probability at least $1-\delta$ we have for each $i\in\{0,\ldots,2k\}$, at least one $j\leq m$ has $\frac{i}{2k+1}<x_j<\frac{i+1}{2k+1}$, and no $j\leq m$ has $x_j=\frac{i}{2k+1}$; in this case, $\hat{\mathcal{C}}_{S_m}$ is constructed as follows; for each $i\in\{1,\ldots,2k\}$, we include in $\hat{\mathcal{C}}_{S_m}$ the point (x_j,y_j) with largest x_j less than $\frac{i}{2k+1}$ and the point (x_j,y_j) with smallest x_j greater than $\frac{i}{2k+1}$. The number of points in this set $\hat{\mathcal{C}}_{S_m}$ is at most $4k$. Therefore, for any $m\in\mathbb{N}$, we have $\mathcal{B}_{\hat{n}}(m,\delta)\leq\min\left\{m,\max\left\{\left\lceil(2k+1)\ln\left(\frac{2k+1}{\delta}\right)\right\rceil,4k\right\}\right\}$. In particular, noting that $d=2k$ here, we have that for $\varepsilon<1/k$, the bound on $\Lambda(\varepsilon,\delta)$ in Lemma 30

has a $\tilde{\Theta}(k^2)$ dependence on k, while the upper bound on $\Lambda(\varepsilon,\delta)$ in Theorem 10 has only a $\tilde{\Theta}(k)$ dependence on k, which matches the lower bound in Theorem 10 (up to logarithmic factors).

Aside from the disagreement coefficient, the other technique in the existing literature for bounding the label complexity of CAL is due to El-Yaniv and Wiener (2010, 2012), based on a quantity they call the *characterizing set complexity*, denoted $\gamma(\mathcal{F},\hat{n}(S_m))$. Formally, for $n\in\mathbb{N}$, let $\gamma(\mathcal{F},n)$ denote the VC dimension of the collection of sets $\{\mathrm{DIS}(\mathrm{VS}_{\mathcal{F},S}) : S\in(\mathcal{X}\times\mathcal{Y})^n\}$. Then El-Yaniv and Wiener (2012) prove the following bound, for a universal constant $c\in(0,\infty)$.[11]

$$\Lambda(\varepsilon,\delta)\le c\left(\max_{m\le M(\varepsilon,\delta/2)}\gamma(\mathcal{F},\mathcal{B}_{\hat{n}}(m,\delta))\ln\left(\frac{em}{\gamma(\mathcal{F},\mathcal{B}_{\hat{n}}(m,\delta))}\right)+\ln\left(\frac{\log_2(2M(\varepsilon,\delta/2))}{\delta}\right)\right)\log_2(2M(\varepsilon,\delta/2)). \quad (6)$$

We can immediately note that $\gamma(\mathcal{F},\mathcal{B}_{\hat{n}}(m,\delta))\ge\mathcal{B}_{\hat{n}}(m,\delta)-1$; specifically, for any $S\in(\mathcal{X}\times\mathcal{Y})^m$, letting $\{(x_{i_1},y_{i_1}),\ldots,(x_{i_{\hat{n}(S_m)}},y_{i_{\hat{n}(S_m)}})\}=\hat{\mathcal{C}}_S$, we have that $\{x_{i_2},\ldots,x_{i_{\hat{n}(S_m)}}\}$ is shattered by $\{\mathrm{DIS}(\mathrm{VS}_{\mathcal{F},S'}) : S'\in(\mathcal{X}\times\mathcal{Y})^{\hat{n}(S_m)}\}$, since letting S' be any subset of $\{(x_{i_2},y_{i_2}),\ldots,(x_{i_{\hat{n}(S_m)}},y_{i_{\hat{n}(S_m)}})\}$ (filling in the remaining elements as copies of (x_{i_1},y_{i_1}) to make S' of size $\hat{n}(S_m)$),

$$\{(x_{i_2},y_{i_2}),\ldots,(x_{i_{\hat{n}(S_m)}},y_{i_{\hat{n}(S_m)}})\}\cap(\mathrm{DIS}(\mathrm{VS}_{\mathcal{F},S'})\times\mathcal{Y})=\{(x_{i_2},y_{i_2}),\ldots,(x_{i_{\hat{n}(S_m)}},y_{i_{\hat{n}(S_m)}})\}\setminus S',$$

since otherwise, the (x_{i_j},y_{i_j}) in $\{(x_{i_2},y_{i_2}),\ldots,(x_{i_{\hat{n}(S_m)}},y_{i_{\hat{n}(S_m)}})\}\setminus S'$ not in $\mathrm{DIS}(\mathrm{VS}_{\mathcal{F},S'})\times\mathcal{Y}$ would have $x_{i_j}\notin\mathrm{DIS}(\mathrm{VS}_{\mathcal{F},\hat{\mathcal{C}}_S\setminus\{(x_{i_j},y_{i_j})\}})$, so that $\mathrm{VS}_{\mathcal{F},\hat{\mathcal{C}}_S\setminus\{(x_{i_j},y_{i_j})\}}=\mathrm{VS}_{\mathcal{F},\hat{\mathcal{C}}_S}=\mathrm{VS}_{\mathcal{F},S}$, contradicting minimality of $\hat{\mathcal{C}}_S$. Therefore, $\gamma(\mathcal{F},\hat{n}(S_m))\ge\hat{n}(S_m)-1$. Then noting that $\gamma(\mathcal{F},n)$ is monotonic in n, we find that $\gamma(\mathcal{F},\mathcal{B}_{\hat{n}}(m,\delta))$ is a minimal $1-\delta$ confidence bound on $\gamma(\mathcal{F},\hat{n}(S_m))$, which implies $\gamma(\mathcal{F},\mathcal{B}_{\hat{n}}(m,\delta))\ge\mathcal{B}_{\hat{n}}(m,\delta)-1$.

One can also give examples where the gap between $\mathcal{B}_{\hat{n}}(m,\delta)$ and $\gamma(\mathcal{F},\mathcal{B}_{\hat{n}}(m,\delta)$ is large, for instance where $\gamma(\mathcal{F},\mathcal{B}_{\hat{n}}(m,\delta))\ge d$ while $\mathcal{B}_{\hat{n}}(m,\delta)=2$ for large m. For instance, consider $\mathcal{X}$ that has d points $w_1,\ldots,w_d$ and 2^{d+1} additional points x_I and z_I indexed by the sets $I\subseteq\{1,\ldots,d\}$, and say $\mathcal{F}$ is the space of classifiers $\{h_J : J\subseteq\{1,\ldots,d\}\}$, where for each $J\subseteq\{1,\ldots,d\}$, $\{x : h_J(x)=+1\}=\{w_i : i\in J\}\cup\{x_I : I\subseteq J\}\cup\{z_I : I\subseteq\{1,\ldots,d\}\setminus J\}$; in particular, the classification on $w_1,\ldots,w_d$ determines the classification on the remaining 2^{d+1} points, and $\{w_1,\ldots,w_d\}$ is shatterable, so that $|\mathcal{F}|=2^d$, and the VC dimension of $\mathcal{F}$ is d. Let P be a distribution that has a uniform marginal distribution over the $2^{d+1}+d$ points in $\mathcal{X}$, and satisfies the realizable case assumption (i.e., $\mathbb{P}(Y=f^*(X)|X)=1$, for some $f^*\in\mathcal{F}$). For any integer $m\ge(2^{d+1}+d)\ln(2/\delta)$, with probability at least $1-\delta$, we have $(x_{\{i\le d:f^*(w_i)=+1\}},+1)\in S_m$ and $(z_{\{i\le d:f^*(w_i)=-1\}},+1)\in S_m$. Since every $h_J\in\mathcal{F}$ with $h_J(x_{\{i\le d:f^*(w_i)=+1\}})=+1$ has $\{i\le d : f^*(w_i)=+1\}\subseteq J=\{i\le d : h_J(w_i)=+1\}$, and every $h_J\in\mathcal{F}$ with $h_J(z_{\{i\le d:f^*(w_i)=-1\}})=+1$ has $\{i\le d : f^*(w_i)=-1\}\subseteq\{1,\ldots,d\}\setminus J=\{i\le d : h_J(w_i)=-1\}$, so that $\{i\le d : f^*(w_i)=+1\}\supseteq\{i\le d : h_J(w_i)=+1\}$, we have that every $h_J\in\mathcal{F}$ with both $h_J(x_{\{i\le d:f^*(w_i)=+1\}})=+1$ and $h_J(z_{\{i\le d:f^*(w_i)=-1\}})=+1$ has $\{i\le d : h_J(w_i)=+1\}=\{i\le d : f^*(w_i)=+1\}$. Since classifiers in $\mathcal{F}$ are completely determined by their classification of $\{w_1,\ldots,w_d\}$, this implies $h_J=f^*$. Therefore, letting $\hat{\mathcal{C}}_{S_m}=\{(x_{\{i\le d:f^*(w_i)=+1\}},+1),(z_{\{i\le d:f^*(w_i)=-1\}},+1)\}$, we have $\mathrm{VS}_{\mathcal{F},\hat{\mathcal{C}}_{S_m}}=\mathrm{VS}_{\mathcal{F},S_m}$, so that $\hat{n}(S_m)\le 2$ (in

11. This result can be derived from their Theorem 15 via reasoning analogous to the derivation of Theorem 10 from Lemma 8 above.

fact, one can easily show $\hat{n}(S_m) = 2$ in this case). Thus, for large m, $\mathcal{B}_{\hat{n}}(m,\delta) \leq 2$. However, for any $I \subseteq \{1,\ldots,d\}$, letting $S = \{(x_{\{1,\ldots,d\}\setminus I}, +1)\}$, we have $h_{\{1,\ldots,d\}\setminus I} \in \mathrm{VS}_{\mathcal{F},S}$, every $h \in \mathrm{VS}_{\mathcal{F},S}$ has $h(w_i) = +1$ for every $i \in \{1,\ldots,d\} \setminus I$, and every $i \in I$ has $h_{(\{1,\ldots,d\}\setminus I)\cup\{i\}} \in \mathrm{VS}_{\mathcal{F},S}$, so that $\mathrm{DIS}(\mathrm{VS}_{\mathcal{F},S}) \cap \{w_1,\ldots,w_d\} = \{w_i : i \in I\}$; therefore, the VC dimension of $\{\mathrm{DIS}(\mathrm{VS}_{\mathcal{F},\{x\}}) : x \in \mathcal{X}\}$ is at least d: that is, $\gamma(\mathcal{F},1) \geq d$. Since we have $\hat{n}(S_m) \geq 1$ whenever S_m contains any point other than $x_{\{\}}$ and $z_{\{\}}$, and this happens with probability at least $1 - (2/(2^{d+1}+d))^m \geq 1 - \delta > \delta$ (when $\delta < 1/2$), this implies we have $\gamma(\mathcal{F},\hat{n}(S_m)) \geq \gamma(\mathcal{F},1) \geq d$ with probability greater than δ, which (by monotonicity of $\gamma(\mathcal{F},\cdot)$) implies $\gamma(\mathcal{F},\mathcal{B}_{\hat{n}}(m,\delta)) \geq d$.

This is not quite strong enough to show a gap between (6) and Theorem 10, since the bounds in Theorem 10 require us to *maximize* over the value of m, which would therefore also include values $\mathcal{B}_{\hat{n}}(m,\delta)$ for $m < (2^{d+1}+d)\ln(2/\delta)$. To exhibit a gap between these bounds, we can simply redefine the marginal distribution of P over $\mathcal{X}$ to have $P(\{w_1\} \times \mathcal{Y}) = 1$. Note that with this distribution, $x_i = w_1$ for all i, with probability 1, so that we clearly have $\hat{n}(S_m) = 1$ almost surely, and hence $\mathcal{B}_{\hat{n}}(m,\delta) = 1$ for all m. As argued above, we have $\gamma(\mathcal{F},1) \geq d$ for this space. Therefore, $\max_{m\leq M} \gamma(\mathcal{F},\mathcal{B}_{\hat{n}}(m,\delta)) \geq d$, while $\max_{m\leq M} \mathcal{B}_{\hat{n}}(m,\delta) \leq 1$, for all $M \in \mathbb{N}$. However, note that unlike the example constructed above for the disagreement coefficient, the gap in this example could potentially be eliminated by replacing the distribution-free quantity $\gamma(\mathcal{F},n)$ with a distribution-dependent complexity measure (e.g., an annealed VC entropy or a bracketing number for $\{\mathrm{DIS}(\mathrm{VS}_{\mathcal{F},S}) : S \in (\mathcal{X}\times\mathcal{Y})^n\}$).

References

K. S. Alexander. Rates of growth and sample moduli for weighted empirical processes indexed by sets. *Probability Theory and Related Fields*, 75:379–423, 1987.

M. Anthony and P.L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999.

P. Auer and R. Ortner. A new PAC bound for intersection-closed concept classes. In *Proceedings of the* 17th *Conference on Learning Theory*, 2004.

M.-F. Balcan and P. M. Long. Active and passive learning of linear separators under log-concave distributions. In *Proceedings of the* 26th *Conference on Learning Theory*, 2013.

M.-F. Balcan, A. Beygelzimer, and J. Langford. Agnostic active learning. In *Proceedings of the* 23rd *International Conference on Machine Learning*, 2006.

M.-F. Balcan, A. Broder, and T. Zhang. Margin based active learning. In *Proceedings of the* 20th *Conference on Learning Theory*, 2007.

M.-F. Balcan, A. Beygelzimer, and J. Langford. Agnostic active learning. *Journal of Computer and System Sciences*, 75(1):78–89, 2009.

M.-F. Balcan, S. Hanneke, and J. Wortman Vaughan. The true sample complexity of active learning. *Machine Learning*, 80(2–3):111–139, 2010.

A. Beygelzimer, S. Dasgupta, and J. Langford. Importance weighted active learning. In *Proceedings of the* 26th *International Conference on Machine Learning*, 2009.

A. Beygelzimer, D. Hsu, J. Langford, and T. Zhang. Agnostic active learning without constraints. In *Advances in Neural Information Processing Systems 23*, 2010.

D. Cohn, L. Atlas, and R. Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994.

S. Dasgupta, D. Hsu, and C. Monteleoni. A general agnostic active learning algorithm. In *Advances in Neural Information Processing Systems 20*, 2007.

R. El-Yaniv and Y. Wiener. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11:1605–1641, 2010.

R. El-Yaniv and Y. Wiener. Active learning via perfect selective classification. *Journal of Machine Learning Research*, 13:255–279, 2012.

E. Friedman. Active learning for smooth problems. In *Proceedings of the* 22nd *Conference on Learning Theory*, 2009.

E. Giné and V. Koltchinskii. Concentration inequalities and asymptotic results for ratio type empirical processes. *The Annals of Probability*, 34(3):1143–1216, 2006.

S. Goldman and M. Kearns. On the complexity of teaching. *Journal of Computer and System Sciences*, 50:20–31, 1995.

S. Hanneke. A bound on the label complexity of agnostic active learning. In *Proceedings of the* 24th *International Conference on Machine Learning*, 2007a.

S. Hanneke. Teaching dimension and the complexity of active learning. In *Proceedings of the* 20th *Conference on Learning Theory*, 2007b.

S. Hanneke. *Theoretical Foundations of Active Learning*. PhD thesis, Carnegie Mellon University, 2009.

S. Hanneke. Rates of convergence in active learning. *The Annals of Statistics*, 39(1):333–361, 2011.

S. Hanneke. Activized learning: Transforming passive to active with improved label complexity. *Journal of Machine Learning Research*, 13(5):1469–1587, 2012.

S. Hanneke. Theory of Active Learning. `http://www.stevehanneke.com`, 2014.

S. Hanneke and L. Yang. Surrogate losses in passive and active learning. *arXiv:1207.3772*, 2012.

T. Hegedüs. Generalized teaching dimensions and the query complexity of learning. In *Proceedings of the* 8th *Conference on Computational Learning Theory*, 1995.

L. Hellerstein, K. Pillaipakkamnatt, V. Raghavan, and D. Wilkins. How many queries are needed to learn? *Journal of the Association for Computing Machinery*, 43(5):840–862, 1996.

R. Herbrich. *Learning Kernel Classifiers*. The MIT Press. Cambridge, MA, 2002.

D. Hsu. *Algorithms for Active Learning*. PhD thesis, Department of Computer Science and Engineering, School of Engineering, University of California, San Diego, 2010.

A. N. Kolmogorov and S. V. Fomin. *Introductory Real Analysis*. Dover, 1975.

V. Koltchinskii. Rademacher complexities and bounding the excess risk in active learning. *Journal of Machine Learning Research*, 11:2457–2485, 2010.

S. R. Kulkarni, S. K. Mitter, and J. N. Tsitsiklis. Active learning using arbitrary binary valued queries. *Machine Learning*, 11:23–35, 1993.

N. Littlestone and M. Warmuth. Relating data compression and learnability. *Unpublished manuscript*, 1986.

P. M. Long. On the sample complexity of PAC learning halfspaces against the uniform distribution. *IEEE Transactions on Neural Networks*, 6(6):1556–1559, 1995.

E. Mammen and A.B. Tsybakov. Smooth discrimination analysis. *The Annals of Statistics*, 27: 1808–1829, 1999.

P. Massart and É. Nédélec. Risk bounds for statistical learning. *The Annals of Statistics*, 34(5): 2326–2366, 2006.

T. Mitchell. Version spaces: A candidate elimination approach to rule learning. In *Proceedings of the* 5^{th} *International Joint Conference on Artificial Intelligence*, 1977.

V. Vapnik. *Estimation of Dependencies Based on Empirical Data*. Springer-Verlag, New York, 1982.

V. Vapnik. *Statistical Learning Theory*. Wiley Interscience, New York, 1998.

V. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16:264–280, 1971.

L. Wang. Smoothness, disagreement coefficient, and the label complexity of agnostic active learning. *Journal of Machine Learning Research*, 12:2269–2292, 2011.

Y. Wiener. *Theoretical Foundations of Selective Prediction*. PhD thesis, the Technion — Israel Institute of Technology, 2013.

Y. Wiener and R. El-Yaniv. Pointwise tracking the optimal regression function. In *Advances in Neural Information Processing Systems 25*, 2012.

Y. Wiener and R. El-Yaniv. Agnostic pointwise-competitive selective classification. *Journal of Machine Learning Research*, to appear, 2015.

Journal of Machine Learning Research 16 (2015) 747-773 Submitted 10/13; Revised 7/14; Published 4/15

Response-Based Approachability with Applications to Generalized No-Regret Problems

Andrey Bernstein ANDREY.BERNSTEIN@EPFL.CH
School of Computer and Communication Sciences
EPFL—École Polytechnique Fédérale de Lausanne
Lausanne CH-1015, Switzerland

Nahum Shimkin SHIMKIN@EE.TECHNION.AC.IL
Department of Electrical Engineering
Technion—Israel Institute of Technology
Haifa 32000, Israel

Editor: Alexander Rakhlin

Abstract

Blackwell's theory of approachability provides fundamental results for repeated games with vector-valued payoffs, which have been usefully applied in the theory of learning in games, and in devising online learning algorithms in the adversarial setup. A target set S is approachable by a player (the agent) in such a game if he can ensure that the average payoff vector converges to S, no matter what the opponent does. Blackwell provided two equivalent conditions for a convex set to be approachable. Standard approachability algorithms rely on the primal condition, which is a geometric separation condition, and essentially require to compute at each stage a projection direction from a certain point to S. Here we introduce an approachability algorithm that relies on Blackwell's *dual* condition, which requires the agent to have a feasible *response* to each mixed action of the opponent, namely a mixed action such that the expected payoff vector belongs to S. Thus, rather than projections, the proposed algorithm relies on computing the response to a certain action of the opponent at each stage. We demonstrate the utility of the proposed approach by applying it to certain generalizations of the classical regret minimization problem, which incorporate side constraints, reward-to-cost criteria, and so-called global cost functions. In these extensions, computation of the projection is generally complex while the response is readily obtainable.

Keywords: approachability, no-regret algorithms

1. Introduction

Consider a repeated matrix game with *vector-valued* rewards that is played by two players, the *agent* and the *opponent*, where the latter may stand for an arbitrarily-varying learning environment. For each pair of simultaneous actions a and b of the agent and the opponent in the one-stage game, a reward vector $r(a,b) \in \mathbb{R}^\ell$, $\ell \geq 1$, is obtained. In the approachability problem formulated in (Blackwell, 1956), the agent's goal is to ensure that the long-term average reward vector *approaches* a given target set S, namely converges to S almost surely in the point-to-set distance. If that convergence can be guaranteed irrespectively of the opponent's strategy, the set S is said to be *approachable*, and the strategy of the agent

that satisfies this property is an approachability strategy (or algorithm) for S. Refinements and extensions of Blackwell's results have been considered, among others, in Vieille (1992); Shimkin and Shwartz (1993); Hart and Mas-Colell (2001); Spinat (2002); Lehrer (2002); Lehrer and Solan (2009); Abernethy et al. (2011).

Blackwell's approachability results have been broadly used in the theoretical work on learning in games, encompassing equilibrium analysis in repeated games with incomplete information (Aumann and Maschler, 1995), calibrated forecasting (Foster, 1999), and convergence to correlated equilibria (Hart and Mas-Colell, 2000). An application of approachability to multi-criteria reinforcement learning was considered in Mannor and Shimkin (2004). The earliest application, however, concerned the notion of *no-regret strategies*, that was introduced in Hannan (1957). Even before Hannan's paper appeared in print, it was shown in Blackwell (1954) that regret minimization can be formulated as a particular approachability problem, leading to an elegant no-regret strategy. More recently, approachability was used in Rustichini (1999) to establish a no-regret result for games with imperfect monitoring, and Hart and Mas-Colell (2001) proposed an alternative approachability formulation of the no-regret problem (see Section 5 for more details). An overview of approachability and no-regret in the context of learning in games can be found in Fudenberg and Levine (1998) and Young (2004), while Cesa-Bianchi and Lugosi (2006) highlights the connection with the modern theory of on-line learning and prediction algorithms. The recent article Perchet (2014) reviews the inter-relations between approachability, regret minimization and calibration.

Standard approachability algorithms require, at each stage of the game, the computation of the direction from the current average reward vector to a closest point in the target set S. This is implied by Blackwell's *primal* geometric separation condition, which is a sufficient condition for approachability of a target set. For *convex* sets, this step is equivalent to computing the *projection direction* of the average reward onto S. In this paper, we introduce an approachability algorithm that avoids this projection computation step. Instead, the algorithm relies on the availability of a *response map*, that assigns to each mixed action q of the opponent a mixed action p of the agent so that $r(p, q)$, the expected reward vector under these two mixed actions, is in S. Existence of such a map is based on the Blackwell's *dual* condition, which is also a necessary and sufficient condition for approachability of a convex target set.

The idea of defining an approachable set in terms of a general response map appears in Lehrer and Solan (2007), in the context of internal no-regret strategies. An explicit approachability algorithm which is based on computing the response to *calibrated forecasts* of the opponent's actions has been proposed in Perchet (2009), and further analyzed in Bernstein et al. (2014). However, the algorithms in these papers are essentially based on computing calibrated forecasts of the opponent's actions, a task which is computationally hard (Hazan and Kakade, 2012). In contrast, the algorithms proposed in the present paper retain the dimensionality of the single-stage game, similarly to Blackwell's original algorithm. An approachability algorithm that combines the response map with no-regret learning was proposed in Bernstein (2013). The algorithm accommodates some additional adaptive properties, but its temporal convergence rate is $O(n^{-1/4})$ rather than $O(n^{-1/2})$. A similar algorithm was employed in Mannor et al. (2014) to elegantly establish approachability results for unknown games.

Our motivation for the proposed algorithms is mainly derived from certain generalizations of the basic no-regret problem, where the set to be approached is geometrically complicated so that computing the projection direction may be hard, while the response map is explicit by construction. These generalizations include the constrained regret minimization problem (Mannor et al., 2009), regret minimization with global cost functions (Even-Dar et al., 2009), regret minimization in variable duration repeated games (Mannor and Shimkin, 2008), and regret minimization in stochastic game models (Mannor and Shimkin, 2003). In these cases, the computation of a response reduces to computing a *best-response* in the underlying regret minimization problem, and hence can be carried out efficiently. The application of our algorithm to some of these problems is discussed in Section 5 of this paper.

The paper proceeds as follows. In Section 2 we review the approachability framework along with available approachability algorithms. Section 3 presents our basic algorithm and establishes its approachability properties. In Section 4, we provide an interpretation of the proposed algorithm, and examine some variants and extensions. Section 5 presents the application to generalized no-regret problems. We conclude the paper in Section 6.

2. Review of Approachability Theory

Let us start with a brief review of the approachability problem. Consider a repeated two-person matrix game, played between an agent and an arbitrary opponent. The agent chooses its actions from a finite set $\mathcal{A}$, while the opponent chooses its actions from a finite set $\mathcal{B}$. At each step $n = 1, 2, ...$, the agent selects its action $a_n \in \mathcal{A}$, observes the action $b_n \in \mathcal{B}$ chosen by the opponent, and obtains a *vector-valued* reward $R_n = r(a_n, b_n) \in \mathbb{R}^\ell$, where $\ell \geq 1$, and $r : \mathcal{A} \times \mathcal{B} \to \mathbb{R}^\ell$ is a given reward function. The average reward vector obtained by the agent up to time n is then $\bar{R}_n = n^{-1} \sum_{k=1}^n R_k$. A *mixed* action of the agent is a probability vector $p \in \Delta(\mathcal{A})$, where $p(a)$ specifies the probability of choosing action $a \in \mathcal{A}$, and $\Delta(\mathcal{A})$ denotes the set of probability vectors over $\mathcal{A}$. Similarly, $q \in \Delta(\mathcal{B})$ denotes a mixed action of the opponent. Let $\bar{q}_n \in \Delta(\mathcal{B})$ denote the empirical distribution of the opponent's actions at time n, namely

$$\bar{q}_n(b) \triangleq \frac{1}{n} \sum_{k=1}^{n} \mathbb{I}\{b_n = b\}, \quad b \in \mathcal{B},$$

where $\mathbb{I}$ denotes the indicator function. Further define the Euclidean span of the reward vector as

$$\rho \triangleq \max_{a,b,a',b'} \left\| r(a,b) - r(a',b') \right\|, \tag{1}$$

where $\|\cdot\|$ is the Euclidean norm. The inner product between two vectors $v \in \mathbb{R}^\ell$ and $w \in \mathbb{R}^\ell$ is denoted by $v \cdot w$.

In what follows, we use the shorthand notation

$$r(p,q) \triangleq \sum_{a \in \mathcal{A}, b \in \mathcal{B}} p(a) q(b) r(a,b)$$

for the expected reward under mixed actions $p \in \Delta(\mathcal{A})$ and $q \in \Delta(\mathcal{B})$; the distinction between $r(a,b)$ and $r(p,q)$ should be clear from their arguments. We similarly denote

$r(p,b) = \sum_{a\in\mathcal{A}} p(a)r(a,b)$ for the expected reward under mixed action $p \in \Delta(\mathcal{A})$ and pure action $b \in \mathcal{B}$.

Let $h_n \triangleq \{a_1, b_1, ..., a_n, b_n\} \in (\mathcal{A}\times\mathcal{B})^n$ denote the history of the game up to stage n. A *strategy* $\pi = (\pi_n)$ of the agent is a collection of decision rules $\pi_n : (\mathcal{A}\times\mathcal{B})^{n-1} \to \Delta(\mathcal{A})$, $n \geq 1$, where each mapping π_n specifies a mixed action $p_n = \pi_n(h_{n-1})$ for the agent at time n. The agent's pure action a_n is sampled from p_n. Similarly, the opponent's strategy is denoted by $\sigma = (\sigma_n)$, with $\sigma_n : (\mathcal{A}\times\mathcal{B})^{n-1} \to \Delta(\mathcal{B})$. Let $\mathbb{P}^{\pi,\sigma}$ denote the probability measure on $(\mathcal{A}\times\mathcal{B})^\infty$ induced by the strategy pair (π,σ).

Let S be a given target set in the reward space. We may assume that S is closed as approachability of a set and its closure are equivalent.

Definition 1 (Approachable Set) *A closed set $S \subseteq \mathbb{R}^\ell$ is* approachable *by the agent if there exists a strategy π of the agent such that $\bar{R}_n = n^{-1}\sum_{k=1}^n R_k$ converges to S in the Euclidean point-to-set distance $d(\cdot, S)$, almost surely for every strategy σ of the opponent, at a uniform rate over the opponent's strategies. That is, for every $\epsilon > 0$ there exists an integer N such that*

$$\mathbb{P}^{\pi,\sigma}\{\sup_{n\geq N} d(\bar{R}_n, S) \geq \epsilon\} \leq \epsilon$$

for any strategy σ of the opponent.

In the sequel, we will find it convenient to state most of our results in terms of the time averaged *expected* rewards, where expectation is applied only to the agent's mixed actions:

$$\bar{r}_n = \frac{1}{n}\sum_{k=1}^n r_k, \quad \text{where} \;\; r_k = r(p_k, b_k).$$

With these smoothed rewards, the stated convergence results and bounds can be shown to hold *pathwise*, for any possible sequence of the opponent's actions. See, e.g., Theorem 4, which states that $d(\bar{r}_n, S) \leq \frac{\rho}{\sqrt{n}}$ for all n. The corresponding almost sure convergence for the actual average reward $\bar{R}_n$ readily follows using martingale convergence theory. Indeed, observe that

$$d\left(\bar{R}_n, S\right) \leq \left\|\bar{R}_n - \bar{r}_n\right\| + d\left(\bar{r}_n, S\right),$$

where the first normed term is the time average of the vector-valued and uniformly bounded martingale difference sequence $D_k = r(a_k, b_k) - r(p_k, b_k)$. By standard martingale results, this average converges to zero at a uniform rate of $O(n^{-1/2})$.

We proceed to present a formulation of Blackwell's results, which provide a sufficient condition for approachability of general sets, and two sets of necessary and sufficient conditions for approachability of *convex* sets. For any $x \notin S$, let $c(x) \in S$ denote a closest point in S to x. Also, for any $p \in \Delta(\mathcal{A})$, let $T(p) = \{r(p,q) : \; q \in \Delta(\mathcal{B})\}$ denote the set of mean reward vectors that are achievable by the opponent. This evidently coincides with the convex hull of the vectors $\{r(p,b)\}_{b\in\mathcal{B}}$.

Definition 2 (Approachability Conditions)

(i) ***B-sets:*** *A closed set $S \subseteq \mathbb{R}^\ell$ will be called a B-set if for every $x \notin S$ there exists a mixed action $p^* = p^*(x) \in \Delta(\mathcal{A})$ and a closest point $c(x) \in S$ such that the hyperplane through $c(x)$ perpendicular to the line segment x-$c(x)$, separates x from $T(p^*)$.*

(ii) ***D-sets:*** *A closed set $S \subseteq \mathbb{R}^\ell$ will be called a D-set if for every $q \in \Delta(\mathcal{B})$ there exists a mixed action $p \in \Delta(\mathcal{A})$ so that $r(p,q) \in S$. We shall refer to such p as a* response *(or S-response) of the agent to q.*

Theorem 3 (Blackwell, 1956)

(i) ***Primal Condition and Algorithm.*** *A B-set is approachable, by using at stage n the mixed action $p^*(\bar{r}_{n-1})$ whenever $\bar{r}_{n-1} \notin S$. If $\bar{r}_{n-1} \in S$, an arbitrary action can be used.*

(ii) ***Dual Condition.*** *A closed set S is approachable only if it is a D-set.*

(iii) ***Convex Sets.*** *Let S be a closed convex set. Then, the following statements are equivalent: (a) S is approachable, (b) S is a B-set, (c) S is a D-set.*

We note that the approachability algorithm in Theorem 3(i) remains valid if $\bar{r}_{n-1}$ in the primal condition is replaced by $\bar{R}_{n-1}$. Blackwell's algorithm was generalized in Hart and Mas-Colell (2001) to a class of approachability algorithms, where the required steering directions are generated as gradients of a suitable potential function (rather than Euclidean projections). An alternative construction was recently proposed in Abernethy et al. (2011), where the steering directions are generated through a no-regret algorithm. Finally, as already mentioned, calibration-based approachability algorithms were considered in Perchet (2009) and Bernstein et al. (2014).

3. Response-Based Approachability

In this section we present our basic response-based algorithm, and establish its convergence properties. In the remainder of the paper, we shall assume that the target set S satisfies the following assumption.

Assumption 1 *The set S is a closed, convex and approachable set.*

It follows by Theorem 3 that S is a D-set, so that for all $q \in \Delta(\mathcal{B})$ there exists an S-response $p \in \Delta(\mathcal{A})$ such that $r(p,q) \in S$. It is further assumed that the agent can compute a response to any q.

We note that in some cases of interest, including those discussed in Section 5, the target S may itself be defined through an appropriate response map. Suppose that for each $q \in \Delta(\mathcal{B})$, we are given a mixed action $p^*(q) \in \Delta(\mathcal{A})$, devised so that $r(p^*(q),q)$ satisfies some desired properties. Then the convex hull $S = \mathrm{conv}\{r(p^*(q),q),\, q \in \Delta(\mathcal{B})\}$ is a convex D-set by construction, hence approachable.

The proposed approachability strategy is presented in Algorithm 1. The general idea is as follows. At each stage n of the algorithm, a *steering vector* $\lambda_{n-1} = \bar{r}^*_{n-1} - \bar{r}_{n-1}$ is computed as the difference between the current average reward and the average of a certain sequence of *target points* r^*_k in S. The target point r^*_n is computed as $r(p^*_n, q^*_n)$, where p^*_n is chosen as an S-response to a certain fictitious action q^*_n of the opponent. Both p_n (the actual mixed action of the agent) and q^*_n are computed in step 3 of the algorithm, as the optimal strategies in the scalar game obtained by projecting the payoff vectors in the direction of

Algorithm 1 Response-Based Approachability

Initialization: At time step $n = 1$, use arbitrary mixed action p_1 and set an arbitrary target point $r_1^* \in S$.

At time step $n = 2, 3, ...$:

1. Set an approachability direction

$$\lambda_{n-1} = \bar{r}^*_{n-1} - \bar{r}_{n-1},$$

where

$$\bar{r}_{n-1} = \frac{1}{n-1}\sum_{k=1}^{n-1} r(p_k, b_k), \qquad \bar{r}^*_{n-1} = \frac{1}{n-1}\sum_{k=1}^{n-1} r^*_k$$

are, respectively, the average (smoothed) reward vector and the average target point.

2. Solve the zero-sum matrix game with payoff matrix defined by $r(a, b)$ projected in the direction λ_{n-1}. Namely, find the equilibrium strategies p_n and q_n^* that satisfy

$$p_n \in \operatorname*{argmax}_{p\in\Delta(\mathcal{A})} \min_{q\in\Delta(\mathcal{B})} \lambda_{n-1} \cdot r(p, q), \tag{2}$$

$$q_n^* \in \operatorname*{argmin}_{q\in\Delta(\mathcal{B})} \max_{p\in\Delta(\mathcal{A})} \lambda_{n-1} \cdot r(p, q), \tag{3}$$

3. Choose action a_n according to p_n.

4. Pick p_n^* so that $r(p_n^*, q_n^*) \in S$, and set the target point $r_n^* = r(p_n^*, q_n^*)$.

λ_{n-1}. As shown in the proof, and further elaborated in Subsection 4.1, this choice implies the convergence of the difference $\lambda_n = \bar{r}^*_n - \bar{r}_n$ to 0. Since $\bar{r}^*_n \in S$ by construction, this in turn implies convergence of $\bar{r}_n$ to S.

We may now present our main convergence result and its proof, followed by some additional comments on the algorithm. Recall that ρ is reward span as defined in (1).

Theorem 4 *Let Assumption 1 hold, and suppose that the agent follows the strategy specified in Algorithm 1. Then*

$$d\left(\bar{r}_n, S\right) \le \|\lambda_n\| \le \frac{\rho}{\sqrt{n}}, \quad n \ge 1, \tag{4}$$

for any strategy of the opponent.

The proof follows from the next result, which also provides more general conditions on the required properties of (p_n, q_n^*, p_n^*).

Proposition 5 *(i) Suppose that at each time step $n \ge 1$, the agent chooses the triple (p_n, q_n^*, p_n^*) so that*

$$\lambda_{n-1} \cdot (r(p_n, b) - r(p_n^*, q_n^*)) \ge 0, \quad \forall b \in \mathcal{B}, \tag{5}$$

and sets $r_n^* = r(p_n^*, q_n^*)$. Then $\|\lambda_n\| \le \frac{\rho}{\sqrt{n}}$ for $n \ge 1$.

(ii) If, in addition, p_n^* is chosen as an S-response to q_n^*, so that $r_n^* = r(p_n^*, q_n^*) \in S$, then

$$d(\bar{r}_n, S) \leq \|\lambda_n\| \leq \frac{\rho}{\sqrt{n}}, \quad n \geq 1, \tag{6}$$

Proof We first observe that

$$n^2 \|\lambda_n\|^2 \leq (n-1)^2 \|\lambda_{n-1}\|^2 + 2(n-1)\lambda_{n-1} \cdot (r_n^* - r_n) + \rho^2, \tag{7}$$

for any $n \geq 1$. Indeed,

$$\begin{aligned} \|\bar{r}_n^* - \bar{r}_n\|^2 &= \left\| \frac{n-1}{n} \left(\bar{r}_{n-1}^* - \bar{r}_{n-1}\right) + \frac{1}{n}\left(r_n^* - r_n\right) \right\|^2 \\ &= \left(\frac{n-1}{n}\right)^2 \|\lambda_{n-1}\|^2 + \frac{1}{n^2}\|r_n^* - r_n\|^2 + 2\frac{n-1}{n^2}\lambda_{n-1} \cdot (r_n^* - r_n) \\ &\leq \left(\frac{n-1}{n}\right)^2 \|\lambda_{n-1}\|^2 + \frac{\rho^2}{n^2} + 2\frac{n-1}{n^2}\lambda_{n-1} \cdot (r_n^* - r_n)\,. \end{aligned}$$

Now, under condition (5),

$$\lambda_{n-1} \cdot (r_n^* - r_n) = \lambda_{n-1} \cdot (r(p_n^*, q_n^*) - r(p_n, b_n)) \leq 0.$$

Hence, by (7),

$$n^2 \|\lambda_n\|^2 \leq (n-1)^2 \|\lambda_{n-1}\|^2 + \rho^2, \quad n \geq 1.$$

Applying this inequality recursively, we obtain that $n^2 \|\lambda_n\|^2 \leq n\rho^2$, or $\|\lambda_n\|^2 \leq \rho^2/n$, as claimed in part (i). Part (ii) now follows since $r_n^* \in S$ implies that $\bar{r}_n^* \in S$ (by convexity of S), hence

$$d(\bar{r}_n, S) \leq \|\bar{r}_n - \bar{r}_n^*\| = \|\lambda_n\|\,.$$

■

Proof [Theorem 4] It only remains to show that the choice of (p_n, q_n^*) in equations (2)-(3) implies the required inequality in (5). Indeed, under (2) and (3) we have that

$$\begin{aligned} \lambda_{n-1} \cdot r(p_n, b_n) &\geq \max_{p \in \Delta(\mathcal{A})} \min_{q \in \Delta(\mathcal{B})} \lambda_{n-1} \cdot r(p, q) \\ &= \min_{q \in \Delta(\mathcal{B})} \max_{p \in \Delta(\mathcal{A})} \lambda_{n-1} \cdot r(p, q) \\ &\triangleq \max_{p \in \Delta(\mathcal{A})} \lambda_{n-1} \cdot r(p, q_n^*), \end{aligned}$$

where the equality follows by the minimax theorem for matrix games. Therefore, condition (5) is satisfied for any p_n^*, and in particular for the one satisfying $r(p_n^*, q_n^*) \in S$. This concludes the proof of Theorem 4. ■

Additional Comments:

1. Observe that the projection directions in Blackwell's algorithm are replaced, in a sense, by the steering vectors λ_n. These vectors are computed based on the agent's S-responses to a fictitious sequence (q_n^*) of the opponent's mixed actions, which is computed as part of the algorithm.

2. Theorem 4 clearly implies that the set S is approachable with the specified strategy, and provides an explicit rate of convergence. In fact, the result is somewhat stronger as it implies convergence of the average reward vector to $\bar{r}_n^* \in S$. This property will be found useful in Proposition 13 below, where certain properties that do not follow from approachability alone are established for the reward-to-cost maximization problem.

3. A stated in Proposition 5, the condition in (5) on the triplets (p_n, q_n^*, p_n^*) is sufficient to ensure the convergence $\lambda_n \to 0$. Equations (2)-(3) specify a specific choice of (p_n, q_n^*) which satisfies these conditions. This choice is useful as it implies (5) for *any* choice of p_n^*.

4. The computational requirements of Algorithm 1 are as follows. At each time step n, two major computations are needed:

   a. Computing (p_n, q_n^*)—the equilibrium strategies in the zero-sum matrix game with the reward function $\lambda_{n-1} \cdot r(p, q)$. This boils down to the solution of the related primal and dual linear programs, and hence can be done efficiently. Note that, given the vector λ_{n-1}, this computation does not involve the target set S.

   b. Computing the S-response p_n^* to q_n^* and the target point $r_n^* = r(p_n^*, q_n^*)$, which is problem dependent. Specific examples are discussed in Section 5.

4. Interpretation and Extensions

We open this section with an illuminating interpretation of the proposed algorithm in terms of a certain approachability problem in an auxiliary game. We then proceed to present three variants and extensions to the basic algorithm; we note that these are not essential for the remainder of the paper and can be skipped at first reading. While each of these variants is presented separately, they may also be combined when appropriate.

4.1 An Auxiliary Game Interpretation

A central part of Algorithm 1 is the choice of the pair (p_n, q_n^*) so that $\bar{r}_n$ tracks $\bar{r}_n^*$, namely $\lambda_n = \bar{r}_n^* - \bar{r}_n \to 0$ (see Equations (2)-(3) and Proposition 5). If fact, the choice of (p_n, q_n^*) in (2)-(3) can be interpreted as Blackwell's strategy for a specific approachability problem in an auxiliary game, which we define next.

Suppose that at stage n, the agent chooses a *pair* of actions $(a, b^*) \in \mathcal{A} \times \mathcal{B}$ and the opponent chooses a pair of actions $(a^*, b) \in \mathcal{A} \times \mathcal{B}$. The vector payoff function, now denoted by v, is given by

$$v((a, b^*), (a^*, b)) = r(a^*, b^*) - r(a, b),$$

so that

$$V_n = r(a_n^*, b_n^*) - R_n.$$

Consider the single-point target set $S_0 = \{0\} \subset \mathbb{R}^\ell$. This set is clearly convex, and we next show that it is a D-set in the auxiliary game. We need to show that for any $\eta \in \Delta(\mathcal{A} \times \mathcal{B})$ there exists $\mu \in \Delta(\mathcal{A} \times \mathcal{B})$ so that $v(\mu, \eta) \in S_0$, namely $v(\mu, \eta) = 0$. That that end, observe that

$$v(\mu, \eta) = r(p^*, q^*) - r(p, q)$$

where p and q^* are the marginal distributions of μ on $\mathcal{A}$ and $\mathcal{B}$, respectively, while p^* and q are the respective marginal distributions of η. Therefore we obtain $v(\mu,\eta)=0$ by choosing μ with the same marginals as η, for example $\{\mu(a,b)=p(a)q^*(b)\}$ with $p=p^*$ and $q^*=q$. Thus, by Theorem 3, S_0 is approachable.

We may now apply Blackwell's approachability strategy to this auxiliary game. Since S_0 is the origin, the direction from S_0 to the average reward $\bar{v}_{n-1}$ is just the average reward vector itself. Therefore, the primal (geometric separation) condition here is equivalent to

$$\bar{v}_{n-1}\cdot v(\mu,\eta)\leq 0,\quad \forall\,\eta\in\Delta(\mathcal{A}\times\mathcal{B})$$

or

$$\bar{v}_{n-1}\cdot(r(p^*,q^*)-r(p,q))\leq 0,\quad \forall\,p^*\in\Delta(\mathcal{A}),q\in\Delta(\mathcal{B}).$$

Now, a pair (p,q^*) that satisfies this inequality is any pair of equilibrium strategies in the zero-sum game with reward v projected in the direction of $\bar{v}_{n-1}$. That is, for

$$p\in\operatorname*{argmax}_{p\in\Delta(\mathcal{A})}\min_{q\in\Delta(\mathcal{B})}\bar{v}_{n-1}\cdot r(p,q), \tag{8}$$

$$q^*\in\operatorname*{argmin}_{q\in\Delta(\mathcal{B})}\max_{p\in\Delta(\mathcal{A})}\bar{v}_{n-1}\cdot r(p,q), \tag{9}$$

it is easily verified that

$$\bar{v}_{n-1}\cdot r(p^*,q^*)\geq\bar{v}_{n-1}\cdot r(p,q),\quad \forall\,p^*\in\Delta(\mathcal{A}),q\in\Delta(\mathcal{B})$$

as required.

The choice of (p_n,q_n^*) in Equations (2)-(3) follows (8)-(9), with λ_{n-1} replacing $\bar{v}_{n-1}$. We note that the two are not identical, as $\bar{v}_n$ is the temporal average of $V_n=r(a_n^*,b_n^*)-r(a_n,b_n)$ while λ_n is the average of the expected difference $r(p_n^*,q_n^*)-r(p_n,b_n)$; however this does not change the approachability result above, and in fact either can be used. More generally, any approachability algorithm in the auxiliary game can be used to choose the pair (p_n,q_n^*) in Algorithm 1.

We note that in our original problem, the mixed action p_n^* is not chosen by an "opponent" but rather specified as part of Algorithm 1. But since the approachability result above holds for an arbitrary choice of p_n^*, it also holds for this particular one.

We proceed to present some additional variants of our algorithm.

4.2 Idling when S is Reached

Recall that in the original approachability algorithm of Blackwell, an *arbitrary* action a_n can be chosen by the agent whenever $\bar{r}_{n-1}\in S$. This may alleviate the computational burden of the algorithm, and adds another degree of freedom that may be used to optimize other criteria.

Such an arbitrary choice of a_n (or p_n) when the average reward is in S is also possible in our algorithm. However, some care is required in setting the average target point $\bar{r}_n^*$ at these time instances, as otherwise the two terms of the difference $\lambda_n=\bar{r}_n^*-\bar{r}_n$ may drift

apart. As it turns out, $\bar{r}_n^*$ should be reset at these times to $\bar{r}_n$, which leads to the following recursion. Set $\bar{r}_0^* = 0$, and let

$$\bar{r}_n^* = \begin{cases} \frac{n-1}{n}\bar{r}_{n-1}^* + \frac{1}{n}r_n^* & \text{if } \bar{r}_n \notin S \\ \bar{r}_n & \text{if } \bar{r}_n \in S \end{cases} \tag{10}$$

for $n \geq 1$. The definition of λ_n as $\bar{r}_n^* - \bar{r}_n$ is retained, so that it satisfies the modified recursion:

$$\lambda_n = \begin{cases} \frac{n-1}{n}\lambda_{n-1} + \frac{1}{n}(r_n^* - r_n), & \text{if } \bar{r}_n \notin S \\ 0, & \text{if } \bar{r}_n \in S, \end{cases} \tag{11}$$

with $\lambda_0 = 0$. Thus, the steering vector λ_n is reset to 0 whenever the average reward $\bar{r}_n$ is in S. With this modified definition, the convergence properties of the algorithm are retained (with the same rates). The proof can be found in Bernstein and Shimkin (2013).

4.3 Directionally Unbounded Target Sets

In some applications of interest, the target set S may be unbounded in certain directions. It is often natural to define the agent's goal in this way even if the reward function is bounded, as it reflects clearly the agent's desire of obtaining a reward which is as large as possible in these directions.[1] Indeed, this is the case in the approachability formulations of the no-regret problem, where the goal is essentially to make the (scalar) average reward as large as possible in hindsight.

In such cases, the requirement that $\lambda_n = \bar{r}_n^* - \bar{r}_n \to 0$, which is a property of our basic algorithm, may be too strong, and may even be counter-productive. For example, suppose that our goal is to increase the first coordinate of the average reward vector $\bar{r}_n$ as much as possible. In that case, allowing negative values of λ_n in that component makes sense (rather than steering it to 0 by reducing $\bar{r}_n$). We propose here a modification of our algorithm that addresses this issue

Given the (closed and convex) target set $S \subset \mathbb{R}^\ell$, let D_S be the set of vectors $d \in \mathbb{R}^\ell$ such that $d + S \subset S$. It may be seen that D_S is a closed and convex cone, which trivially equals $\{0\}$ if (and only if) S is bounded. We refer to the unit vectors in D_S as directions in which S is unbounded.

Referring to the auxiliary game interpretation of our algorithm in Section 4.1, we may now relax the requirement that λ_n approaches $\{0\}$ to the requirement that λ_n approaches $-D_S$. Indeed, if we maintain $\bar{r}_n^* \in S$ as before, then $\lambda_n \in -D_S$ suffices to verify that $\bar{r}_n = \bar{r}_n^* - \lambda_n \in S$.

We may now apply Blackwell's approachability strategy to the cone D_S in place of the origin. The required modification to the algorithm is simple: replace the steering direction λ_n in (2)-(3) or (5) with the direction from the closest point in $-D_S$ to λ_n:

$$\tilde{\lambda}_n = \lambda_n - \mathrm{Proj}_{-D_S}(\lambda_n)$$

That projection is particularly simple in case S is unbounded along primary coordinates, so that the cone D_S is a quadrant, generated by a collection $e_j, j \in J$ of orthogonal unit

1. Clearly, it is always possible to intersect S with the bounded set of feasible reward vectors without changing its approachability properties. We find it useful here to retain S in its unbounded form.

vectors. In that case, clearly,

$$\mathrm{Proj}_{-D_S}(\lambda) = -\sum_{j\in J}(e_j \cdot \lambda)^- .$$

Thus, the negative components of λ_n in directions (e_j) are nullified.

The modified algorithm admits analogous bounds to those of the basic algorithm, with (4) or (6) replaced by

$$d\left(\bar{r}_n, S\right) \le d(\lambda_n, -D_S) \le \frac{\rho}{\sqrt{n}}, \quad n \ge 1.$$

The proof is identical, and is obtained by replacing λ_n with $\tilde{\lambda}_n = \lambda_n - \mathrm{Proj}_{-D_S}(\lambda_n)$ in all the relations. See Bernstein and Shimkin (2013) for details.

4.4 Using the Actual Rewards

In the basic algorithm of Section 3, the definition of the steering direction λ_n employs the expected rewards $r(p_k, b_k)$ rather than the actual rewards $R_k = r(a_k, b_k)$. We consider here the variant of the algorithm which employs the latter. This is essential in case that the opponent's action b_k is not observed, so that $r(p_k, b_k)$ cannot be computed, while the reward vector R_k is observed directly. It also makes some sense in general since the quantity we are actually interested in is the average reward $\bar{R}_n$, and not its expected version $\bar{r}_n$.

Thus, we replace λ_{n-1} with

$$\tilde{\lambda}_{n-1} = \bar{r}^*_{n-1} - \bar{R}_{n-1}.$$

The rest of the algorithm remains the same as Algorithm 1. We have the following result for this variant.

Theorem 6 *Let Assumption 1 holds. If the agent uses Algorithm 1, with λ_{n-1} replaced by*

$$\tilde{\lambda}_{n-1} = \bar{r}^*_{n-1} - \bar{R}_{n-1},$$

it holds that

$$\lim_{n\to\infty} \|\tilde{\lambda}_n\| = 0,$$

almost surely, for any strategy of the opponent, at a uniform rate of $O(1/\sqrt{n})$ over all strategies of the opponent. More precisely, for every $\epsilon > 0$,

$$\mathbb{P}\left\{\sup_{k\ge n} \|\tilde{\lambda}_k\| \ge \epsilon\right\} \le \frac{2\rho^2}{n\epsilon^2}. \tag{12}$$

Proof First observe that Lemma 7 still holds if $r_n = r(p_n, b_n)$ is replaced with $R_n = r(a_n, b_n)$ throughout. Namely,

$$n^2\|\tilde{\lambda}_n\|^2 \le (n-1)^2\|\tilde{\lambda}_{n-1}\|^2 + 2(n-1)\tilde{\lambda}_{n-1}\cdot(r^*_n - r(a_n, b_n)) + \rho^2, \quad n \ge 1.$$

Let $\{\mathcal{F}_n\}$ denote the filtration induced by the history. We have that

$$\begin{aligned}\mathbb{E}\left[n^2\|\tilde{\lambda}_n\|^2 \;\middle|\; \mathcal{F}_{n-1}\right] &\leq (n-1)^2\|\tilde{\lambda}_{n-1}\|^2 + 2(n-1)\tilde{\lambda}_{n-1}\cdot\mathbb{E}\left[(r_n^* - r(a_n,b_n)) \mid \mathcal{F}_{n-1}\right] + \rho^2 \\ &= (n-1)^2\|\tilde{\lambda}_{n-1}\|^2 + 2(n-1)\tilde{\lambda}_{n-1}\cdot(r_n^* - \mathbb{E}\left[r(a_n,b_n) \mid \mathcal{F}_{n-1}\right]) + \rho^2 \\ &\leq (n-1)^2\|\tilde{\lambda}_{n-1}\|^2 + \rho^2, \qquad (13)\end{aligned}$$

where the equality follows since q_n^* and p_n^* are determined by the history up to time $n-1$ and hence so does $r_n^* = r(p_n^*, q_n^*)$, and the last inequality holds since

$$\tilde{\lambda}_{n-1}\cdot(r_n^* - \mathbb{E}\left[r(a_n,b_n) \mid \mathcal{F}_{n-1}\right]) = \tilde{\lambda}_{n-1}\cdot(r_n^* - r(p_n,b_n)) \leq 0,$$

similarly to the proof of Theorem 4.

From (13) we may deduce the almost sure convergence $\|\tilde{\lambda}_n\|$ to zero, at a rate the depends on ρ only. The argument may follow the original proof of Blackwell's theorem (Blackwell (1956), Theorem 1), or its adaptation in Shimkin and Shwartz (1993, Proposition 4.1) or Mertens et al. (1994, p. 125) which rely on Doob's maximal inequality for supermartingales. In particular, following the latter reference, we obtain the bound stated in (12). ■

5. Applications to Generalized No-Regret Problems

Our response-based approachability algorithm can be usefully applied to several generalized regret minimization problems, for which computation of a projection onto the target set is involved, but a response is readily obtainable. In the next Subsection, we briefly review the basic no-regret problem and its two standard formulations as an approachability problem. In Subsection 5.2 we first outline a generic *generalized no-regret* problem, using a general set-valued goal function, and then specialize the discussion to some specific problems that have been considered in the recent literature, namely constrained regret minimization, reward-to-cost maximization, and the so-called global cost function problem. In each case, we specify the performance obtainable by a suitable approachability algorithm, along with the corresponding response map that is needed in our algorithm. For the reward-to-cost problem, we also derive some performance guarantees that rely on specific properties of the proposed approachability algorithm.

We do not specify convergence rates in this section, but rather focus on asymptotic convergence results. Convergence rates can be derived by referring to our bounds in the previous sections, namely (4) or (12).

5.1 Approachability-Based No-Regret Algorithms

Let us start by reviewing the basic no-regret problem for repeated matrix games, along with its two alternative formulations as an approachability problem by Blackwell (1954) and Hart and Mas-Colell (2001). Consider, as before, an agent that faces an arbitrarily varying environment (the opponent). The repeated game model is the same as above, except that the vector reward function r is replaced by a scalar reward (or utility) function

$u : \mathcal{A} \times \mathcal{B} \to \mathbb{R}$. Let $\bar{U}_n \triangleq n^{-1} \sum_{k=1}^{n} U_k$ denote the average reward by time n, and let

$$u^*(\bar{q}_n) \triangleq \max_{a \in \mathcal{A}} u(a, \bar{q}_n) = \frac{1}{n} \max_{a \in \mathcal{A}} \sum_{k=1}^{n} u(a, b_k) \tag{14}$$

denote the *best reward-in-hindsight* of the agent after observing $b_1, ..., b_n$, which is a *convex* function u^* of the empirical distribution $\bar{q}_n$. Hannan (1957) introduced the following notion of a no-regret strategy:

Definition 7 (No-Regret Algorithm) *A strategy of the agent is termed a no-regret algorithm (or Hannan Consistent) if*

$$\limsup_{n \to \infty} \left(u^*(\bar{q}_n) - \bar{U}_n \right) \leq 0$$

with probability 1, for any strategy of the opponent.

a. Blackwell's No-Regret Algorithm. Following Hannan's seminal paper, Blackwell (1954) used his approachability theorem to elegantly show the existence of regret minimizing strategies. Define the vector-valued rewards $R_n \triangleq (U_n, \mathbf{1}(b_n)) \in \mathbb{R} \times \Delta(\mathcal{B})$, where $\mathbf{1}(b)$ is the probability vector in $\Delta(\mathcal{B})$ supported on b. The corresponding average reward is $\bar{R}_n \triangleq n^{-1} \sum_{k=1}^{n} R_k = \left(\bar{U}_n, \bar{q}_n \right)$. Finally, define the target set

$$S = \{(u, q) \in \mathbb{R} \times \Delta(\mathcal{B}) : \ u \geq u^*(q)\} .$$

This set is a D-set by construction: An S-response to q is given by any $p^* \in \Delta(\mathcal{A})$ that maximizes $u(p, q)$, as $u(p^*, q) = u^*(q)$ implies that $r(p^*, q) = (u(p^*, q), q) \in S$. Also, S is a convex set by the convexity of $u^*(q)$ in q. Hence, by Theorem 3, S is approachable, and by the continuity of $u^*(q)$, an algorithm that approaches S also minimizes the regret in the sense of Definition 7. Application of Blackwell's approachability strategy to the set S therefore results in a no-regret algorithm. We note that the required projection of the average reward vector onto S cannot be defined explicitly in this formulation. However, the computation of the S-response is explicit and straightforward: We just need to solve the original optimization problem $\max_{p \in \Delta(\mathcal{A})} u(p, q)$, which clearly admits a solution in pure actions.

b. Regret Matching. An alternative formulation due to Hart and Mas-Colell (2001) leads to a simple and explicit no-regret algorithm. Let

$$L_n(a') \triangleq \frac{1}{n} \sum_{k=1}^{n} \left(u(a', b_k) - u(a_k, b_k) \right) \tag{15}$$

denote the regret accrued due to not using action a' exclusively up to time n. The no-regret requirement in Definition 7 is now equivalent to $\limsup_{n \to \infty} L_n(a) \leq 0, a \in \mathcal{A}$, a.s. for any strategy of the opponent. This property, in turn, is equivalent to the approachability of the negative orthant $S = (\mathbb{R}_-)^{\mathcal{A}}$ in the game with vector payoff $r = (r_{a'}) \in \mathbb{R}^{\mathcal{A}}$, defined as $r_{a'}(a, b) = u(a', b) - u(a, b)$.

To verify the dual condition, observe that $r_{a'}(p, q) = u(a', q) - u(p, q)$. Choosing $p \in \operatorname{argmax}_p u(p, q)$ clearly ensures $r(p, q) \in S$, hence is an S-response to q (in the sense of

Definition 2(ii)), and S is a D-set. Note that the response here can always be taken as a pure action.

It was shown in Hart and Mas-Colell (2001) that the application of Blackwell's approachability strategy (or some generalizations thereof) to this formulation is simple and leads to explicit no-regret algorithms, namely the so-called *regret matching* algorithm and its variants.

5.2 Generalized No-Regret

Consider a repeated matrix game as before, except that the vector-valued reward $r(a,b)$ is now denoted by $v(a,b) \in \mathbb{R}^K$. Suppose that for each mixed action q of the opponent, the agent defines a *target set* $V^*(q) \subset \mathbb{R}^K$ which is non-empty and closed. Let $V^* : \Delta(\mathcal{B}) \rightrightarrows \mathbb{R}^K$ denote the corresponding set-valued map, which assigns to each q the subset $V^*(q)$. We refer to V^* as the agent's *goal function*. Denote[2] $v_n = v(a_n, b_n)$, $\bar{v}_n = \frac{1}{n}\sum_{k=1}^{n} v_k$.

Definition 8 (Attainability) *A strategy of the agent is said to be no-regret strategy with respect to the set-valued goal function V^* if*

$$\lim_{n\to\infty} d(\bar{v}_n, V^*(\bar{q}_n)) = 0 \quad (a.s),$$

for any strategy of the opponent. If such a strategy exists we say that V^* is attainable by the agent.

The classical no-regret problem is obtained as a special case, with scalar rewards $v(a,b)$ and target set $V^*(q) = \{u \in \mathbb{R} : u \geq v^*(q)\}$, where $v^*(q) \triangleq \max_p u(p,q)$.

Attainability is closely related to approachability of the graph of V^*. Recall that the graph of a set-valued map $V : \Delta(\mathcal{B}) \rightrightarrows \mathbb{R}^K$ is defined as

$$\mathrm{Graph}(V) \triangleq \left\{(v,q) \in \mathbb{R}^K \times \Delta(\mathcal{B}) :\ v \in V(q)\right\}.$$

(For this and other properties of set-valued maps see, e.g., Aubin and Frankowska, 1990 or Rockafellar and Wets, 1997, Chapter 5.) It is easily seen that attainability of V^* implies approachability of $\mathrm{Graph}(V)$, in the game with augmented vector rewards $r(p,q) = (v(p,q), q)$. The converse is also true under a continuity requirement.

Lemma 9 *Let $V : q \mapsto V^*(q) \cap \mathcal{V}_0$ denote the restriction of V^* to the compact set $\mathcal{V}_0 = \mathrm{conv}\{v(a,b)\}$ of feasible reward vectors. Suppose that V is continuous in the Hausdorff metric. If $\mathrm{Graph}(V^*)$ is approachable in the repeated game with reward vector $r(p,q) = (v(p,q),q)$, then V^* is attainable. Specifically, any approachability strategy for $\mathrm{Graph}(V^*)$ is a no-regret strategy for V^*.*

Proof Clearly, since $\bar{v}_n \in \mathcal{V}_0$, if $\mathrm{Graph}(V^*)$ is approachable then so is $\mathrm{Graph}(V)$, and we may restrict attention to the latter. Recall that the Hausdorff distance $d_{\mathcal{H}}$ between sets X and Y, defined by

$$d_{\mathcal{H}}(X,Y) = \max\{\sup_{x\in X} d(x,Y), \sup_{y\in Y} d(y,X)\},$$

2. For notational convenience, we will not use here the capitalized notation $V_n = v(a_n, b_n)$ to distinguish the latter from $v(p_n, b_n)$, as was done above for r. In fact, v_n can stand for either in the following, depending on whether Algorithm 1 or its variant in Subsection 4.4 is used.

is a metric on the space of non-empty compact subsets of $\mathbb{R}^K$. Now, V may be viewed as a map from the compact set $\Delta(\mathcal{B})$ to the metric space of non-empty compact subsets of $\mathbb{R}^K$ with the Hausdorff metric, and is continuous in that metric by assumption. Hence, by the Heine-Cantor Theorem, V is *uniformly* continuous.

Now, since $S = \text{Graph}(V)$ is approachable, we have (w.p. 1) that $d\left((\bar{v}_n, \bar{q}_n), S\right) \to 0$, implying that

$$\|\bar{v}_n - v_n^*\| \to 0, \quad \|\bar{q}_n - q_n^*\| \to 0,$$

for some sequences $v_n^* \in V(q_n^*)$, $q_n^* \in \Delta(\mathcal{B})$. The uniform continuity of V in the Hausdorff distance $d_{\mathcal{H}}$ then implies that $d_{\mathcal{H}}\left(V(\bar{q}_n), V(q_n^*)\right) \to 0$, hence

$$d(\bar{v}_n, V(\bar{q}_n)) \leq \|\bar{v}_n - v_n^*\| + d_{\mathcal{H}}\left(V(\bar{q}_n), V(q_n^*)\right) \to 0,$$

so that V is attainable by Definition 8. Attainability of V^* now follows since $V(\bar{q}_n) \subseteq V^*(\bar{q}_n)$. ■

We may now formulate a sufficient condition for attainability of a goal function by employing the *dual* condition for approachability of convex sets. Recall that a set-valued map $V : \Delta(\mathcal{B}) \rightrightarrows \mathbb{R}^K$ is called *convex* if its graph $\text{Graph}(V)$ is a convex set. The convex hull $\text{conv}(V)$ of V is the unique set-valued map whose graph is $\text{conv}(\text{Graph}(V)$, the convex hull of $\text{Graph}(V)$. Similarly, the closed convex hull $\overline{\text{co}}(V)$ of V is the unique set-valued map whose graph is the closure of $\text{conv}(\text{Graph}(V))$.

Proposition 10 *Suppose that the set-valued goal function V^* is feasible, in the following sense:*

- *For each mixed action $q \in \Delta(\mathcal{B})$ of the opponent, there exists some mixed action $p = p^*(q)$ of the agent so that $v(p, q) \in V^*(q)$. We refer to $p^*(q)$ as the agent's response to q.*

Denote $V^c = \overline{\text{co}}(V^*)$. Then

(i) The set $\text{Graph}(V^c)$ is approachable by the agent.

(ii) The set-valued goal function V^c is attainable by the agent (in the sense of Definition 8), and any approachability strategy for $\text{Graph}(V^c)$ is a no-regret strategy for V^c.

Proof Let us first redefine V^* as its restriction to the compact set $\mathcal{V}_0$, as in Lemma 9. It is clear that this restricted V^* still satisfies the feasibility requirement of the Proposition, and that establishing the claimed attainability property for the restricted version implies the same for the original one.

Let $V^c = \overline{\text{co}}(V^*)$. We first claim that $\text{Graph}(V^c)$ is approachable. By the assumed feasibility of V^*, for any q there exists p such that $(v(p, q), q) \in S \triangleq \text{Graph}(V^*)$. Therefore $\overline{\text{co}}(S)$ is a convex D-set, which is approachable by Theorem 3. Now, observe that $\overline{\text{co}}(S) = \overline{\text{co}}(\text{Graph}(V^*)) = \text{Graph}(V^c)$ by definition of V^c.

To conclude that V^c is attainable, it remains to verify that it satisfies the continuity requirement in Lemma 9. Observe that $V^c : \Delta(\mathcal{B}) \rightrightarrows \mathcal{V}_0$ is a convex, compact-valued multifunction whose domain is a polytope. By Mačkowiak (2006, Corrolary 2), V^c is lower

semi-continuous.[3] Furthermore, since the graph of $V^c = \overline{\text{co}}(V^*)$ is closed by its definition, V^c is upper-semi-continuous (Rockafellar and Wets, 1997, Theorem 5.7). It follows that V^c is a continuous map. Finally, since standard (Kuratowski) continuity and Hausdorff-continuity are equivalent for compact-valued map (*Ibid.*, 4.40(a)), the required continuity property of V_c follows. This concludes the proof. ■

Proposition 10 implies that a feasible and continuous goal function V^* that is *convex* is attainable. When V^* is not convex, as is often the case in the following examples, we need to resort to its convex relaxation $V^c = \overline{\text{co}}(V^*)$. The suitability of V^c as a goal function needs to be examined for each specific problem.

Proposition 10 asserts also that V^c can be attained by any approachability algorithm applied to the convex set $S = \text{Graph}(V^c)$. However, a projection onto that set as required in the standard approachability algorithms may be hard to compute. This is especially true when V^* itself is non-convex, so that V^c is not explicitly specified. In such cases, the response-based approachability algorithm proposed in this paper offers a convenient alternative, as it only requires to compute at each stage a response $p^*(q)$ of the agent to a mixed action q of the opponent, which is inherent in the definition of V^*.

The resulting generalized no-regret algorithm is presented in Algorithm 2. It is merely an application of Algorithm 1 to the problem of approaching $S = \text{Graph}(V^c)$, with augmented reward vectors $r(p, q) = (u(p, q), q)$.

We next specialize the discussion to certain concrete problems of interest.

5.2.1 Constrained Regret Minimization

The following constrained regret minimization problem was introduced in Mannor et al. (2009). Consider the repeated game model as before, where we are given a scalar reward (or utility) function $u : \mathcal{A} \times \mathcal{B} \to \mathbb{R}$ and a vector-valued cost function $c : \mathcal{A} \times \mathcal{B} \to \mathbb{R}^s$. We are also given a closed and convex set $\Gamma \subseteq \mathbb{R}^s$, the constraint set, which specifies the allowed values for the long-term average cost. A specific case is that of upper bounds on each cost component, that is $\Gamma = \{c \in \mathbb{R}^s : \ c_i \leq \gamma_i, \ i = 1, ..., s\}$ for some given vector $\gamma \in \mathbb{R}^s$. The constraint set is assumed to be *feasible* (or *non-excludable*), in the sense that for every $q \in \Delta(\mathcal{B})$, there exists $p \in \Delta(\mathcal{A})$ such that $c(p, q) \in \Gamma$.

Let $\bar{U}_n \triangleq n^{-1} \sum_{k=1}^n u_k$ and $\bar{C}_n \triangleq n^{-1} \sum_{k=1}^n c_k$ denote, respectively, the average reward and cost by stage n. The agent is required to satisfy the cost constraints, in the sense that $\lim_{n\to\infty} d(\bar{C}_n, \Gamma) = 0$ must hold, irrespectively of the opponent's play. Subject to these constraints, the agent wishes to maximize its average reward $\bar{U}_n$.

We note that a concrete learning application for the constrained regret minimization problem was proposed in Bernstein et al. (2010). There, we considered the on-line problem of merging the output of multiple binary classifiers, with the goal of maximizing the true-positive rate, while keeping the false-positive rate under a given threshold $0 < \gamma < 1$. As shown in that paper, this may be formulated as a constrained regret minimization problem.

3. This is a generalization of the Gale-Klee-Rockfellar Theorem from convex analysis to set-valued maps. The point is of course continuity at the boundary points.

Algorithm 2 Generalized No-Regret Algorithm

Input: The reward function $v : \mathcal{A} \times \mathcal{B} \to \mathbb{R}^K$; a set-valued goal function $V^* : \Delta(\mathcal{B}) \rightrightarrows \mathbb{R}^K$; and for each $q \in \Delta(\mathcal{B})$, a mixed action (or actions) $p \in \Delta(\mathcal{A})$ such that $v(p, q) \in V^*(q)$.

Initialization: At step $n = 1$, apply an arbitrary mixed action p_1, and choose arbitrary values $v_1^* \in \mathbb{R}^K$, $q_1^* \in \Delta(\mathcal{B})$.

At step $n = 2, 3, ...$:

1. Set
$$\lambda_{n-1}^v = \bar{v}_{n-1}^* - \bar{v}_{n-1}, \quad \lambda_{n-1}^q = \bar{q}_{n-1}^* - \bar{q}_{n-1},$$
where
$$(\bar{v}_m^*, \bar{v}_m) = \frac{1}{m}\sum_{k=1}^{m}(v_k^*, v_k), \quad \bar{q}_m^* = \frac{1}{m}\sum_{k=1}^{m} q_k^*, \quad \bar{q}_m = \frac{1}{m}\sum_{k=1}^{m}\mathbb{I}_{\{b_k=\cdot\}},$$
and $v_k = v(p_k, b_k)$ or $v(a_k, b_k)$.
2. Solve the following zero-sum matrix game:
$$p_n \in \operatorname*{argmax}_{p\in\Delta(\mathcal{A})} \min_{q\in\Delta(\mathcal{B})} \left(\lambda_{n-1}^v \cdot v(p, q) + \lambda_{n-1}^q \cdot q\right),$$
$$q_n^* \in \operatorname*{argmin}_{q\in\Delta(\mathcal{B})} \max_{p\in\Delta(\mathcal{A})} \left(\lambda_{n-1}^v \cdot v(p, q) + \lambda_{n-1}^q \cdot q\right).$$
3. Draw an action a_n randomly from p_n.
4. Pick $p_n^* \in \Delta(\mathcal{A})$ such that $v\left(p_n^*, q_n^*\right) \in V^*(q_n^*)$, and set $v_n^* = v(p_n^*, q_n^*)$.

A natural extension of the best-reward-in-hindsight $u^*(q)$ in (14) to the constrained setting is given by
$$u_\Gamma^*(q) \triangleq \max_{p\in\Delta(\mathcal{A})} \{u(p, q) \,:\, c(p, q) \in \Gamma\}. \tag{16}$$
We can now define the target set of the pairs $v = (u, c) \in \mathbb{R}^{1+s}$ in terms of $u_\Gamma^*(q)$ and Γ:
$$V^*(q) \triangleq \left\{v = (u, c) \in \mathbb{R}^{1+s} : u \geq u_\Gamma^*(q), c \in \Gamma\right\}.$$
Note that $u_\Gamma^*(q)$ is *not* convex in general, and consequently $V^*(q)$ is not convex as well. Indeed, it was shown in Mannor et al. (2009) that $V^*(q)$ is not attainable in general. The closed convex hull of $V^*(q)$ may be written as
$$V^c(q) = \left\{(u, c) \in \mathbb{R}^{s+1} :\; u \geq \overline{\text{conv}}\left(u_\Gamma^*\right)(q),\; c \in \Gamma\right\}, \tag{17}$$
where the real-valued function $\overline{\text{conv}}\left(u_\Gamma^*\right)$ is the closure of the lower convex hull of u_Γ^* over $\Delta(\mathcal{A})$.

Two algorithms were proposed in Mannor et al. (2009) for attaining $V^c(q)$. The first is a standard (Blackwell) approachability algorithm for $S = \{(v, q) \,:\, v \in V^c(q)\}$, which

requires the demanding computation of S and the projection directions to S. The second algorithm employs a best-response to calibrated forecasts of the opponent's mixed actions. As mentioned in the introduction, obtaining these forecasts is computationally hard. In contrast, our algorithm mainly requires the computation of the response $p^*(q)$ by solving the maximization problem in (16), which is a convex program. This further reduces to a linear program when the constraints are linear.

Specializing Proposition 10 to this case, we obtain the following result.

Corollary 11 *Consider Algorithm 2 applied to the present model. Thus, the response p_n^* to q_n^* is chosen as any maximizing action in (16) with $q = q_n^*$, and the target point is set to $v_n^* = (u(p_n^*, q_n^*), c(p_n^*, q_n^*))$. Then the goal function V^c is attainable in the sense of Definition 8, which implies that*

$$\liminf_{n\to\infty} \left(\bar{U}_n - \overline{\mathrm{conv}}\,(u_\Gamma^*)\,(\bar{q}_n)\right) \geq 0, \quad \textit{and} \quad \lim_{n\to\infty} d\left(\bar{C}_n, \Gamma\right) = 0 \quad (a.s.)$$

for any strategy of the opponent.

We further note that $V^c(q)$ is unbounded in the direction of its first coordinate u, so that the variant of the algorithm presented in Subsection 4.3 can be applied. In this case, the first coordinate of the steering direction λ_n can be set to zero in $\tilde{\lambda}_n$ whenever it is negative. This corresponds to $\bar{u}_{n-1} \geq \bar{u}_{n-1}^*$, thereby avoiding an unnecessary reduction in $\bar{u}_{n-1}$. Similarly, for a component-wise constraint set of the form $\{c_i \leq \gamma_i\}$, the c_i-coordinate of λ_n may be nullified whenever $[\bar{c}_{n-1}]_i \leq [\bar{c}_{n-1}^*]_i$. The results of Corollary 11 are maintained of course.

5.2.2 Reward-to-Cost Maximization

Consider the repeated game model as before, where the goal of the agent is to maximize the ratio $\bar{U}_n/\bar{C}_n$. Here, $\bar{U}_n$ is, as before, the average of a scalar reward function $u(a,b)$ and $\bar{C}_n$ is the average of a *scalar* and *positive* cost function $c(a,b)$. This problem is mathematically equivalent to regret minimization in repeated games with variable stage duration considered in Mannor and Shimkin (2008) (MS08 for short; in that paper, the cost was specifically taken as the stage duration). Moreover, it can be seen that this problem is a particular case of the global cost function model presented below. However, a direct application of Proposition 10 does not yield a meaningful result in this specific case. We therefore resort to specific analysis which relies on additional properties of our response-based approachability algorithm. This yields a similar bound to that of Proposition 14(ii) below, but without the requirement that G be convex.

Similar bounds to the ones established below were obtained in MS08. The algorithm there was based on playing a best-response to calibrated forecasts of the opponent's mixed actions. The present formulation therefore offers an alternative algorithm which is considerably less demanding computationally.

Denote

$$\rho(a,q) \triangleq \frac{u(a,q)}{c(a,q)}, \quad \rho(p,q) \triangleq \frac{u(p,q)}{c(p,q)}.$$

and let

$$\mathrm{val}(\rho) \triangleq \max_{p\in\Delta(\mathcal{A})} \min_{q\in\Delta(\mathcal{B})} \rho(p,q) = \min_{q\in\Delta(\mathcal{B})} \max_{p\in\Delta(\mathcal{A})} \rho(p,q)$$

(the last equality is proved in MS08; note that $\rho(p,q)$ is not generally concave-convex). As further shown in MS08, $\mathrm{val}(\rho)$ is the value of the zero-sum repeated game with payoffs $\bar{U}_n/\bar{C}_n$, hence serves as a security level for the agent. A natural goal for the agent would be to improve on $\mathrm{val}(\rho)$ whenever the opponent's actions deviate (in terms of their empirical mean) from the minimax optimal strategy.

We next propose an attainable goal function that satisfies this requirement. To that end, let

$$\rho^*(q) \triangleq \max_{p\in\Delta(A)} \rho(p,q)$$

denote the *best ratio-in-hindsight*. Let us apply Algorithm 2, with $v = (u,c)$, and the vector-valued goal function

$$V^*(q) = \left\{ v=(u,c) : \ \frac{u}{c} \geq \rho^*(q) \right\} \tag{18}$$

(observe that $\rho^*(q)$ and $V^*(q)$ are non-convex functions in general). The agent's response is given by any mixed action

$$p^*(q) \in P^*(q) \triangleq \operatorname*{argmax}_{p\in\Delta(\mathcal{A})} \rho(p,q).$$

It is readily verified that the maximum can always be obtained here in pure actions (MS08; see also the proof of Prop. 13 below). Hence, computing the response is trivial in this case.

Denote

$$A^*(q) \triangleq \operatorname*{argmax}_{a\in\mathcal{A}} \rho(a,q),$$

and define the following relaxation of $\rho^*(q)$:

$$\begin{aligned}\rho_1(q) &\triangleq \inf \left\{ \frac{\sum_{j=1}^J u(a_j,q_j)}{\sum_{j=1}^J c(a_j,q_j)} \ : \ J\geq 1,\ q_j\in\Delta(\mathcal{B}),\ \frac{1}{J}\sum_{j=1}^J q_j = q,\ a_j\in A^*(q_j) \right\} \\ &\leq \rho^*(q).\end{aligned} \tag{19}$$

We will show below that ρ_1 is attainable by applying Algorithm 2 to this problem. First, however, we show that ρ_1 never falls below the security level $\mathrm{val}(\rho)$, and is strictly better in typical cases.

Lemma 12

(i) $\rho_1(q) \geq \mathrm{val}(\rho)$ *for all* $q\in\Delta(\mathcal{B})$.

(ii) $\rho_1(q) > \mathrm{val}(\rho)$ *whenever* $\rho^*(q) > \mathrm{val}(\rho)$.

(iii) $\rho_1(q) = \rho^*(q)$ *for the* q*'s that represent pure actions.*

(iv) $\rho_1(q)$ *is a continuous function of* q.

Proof To prove this Lemma, we first derive a more convenient expression for $\rho_1(q)$. For $a \in \mathcal{A}$, let

$$Q_a \triangleq \{q \in \Delta(\mathcal{B}) \,:\, a \in A^*(q)\}$$

denote the (closed) set of mixed actions to which a is a best-response action. Observe that for given J, $q_1, ..., q_J$ and $a_j \in A^*(q_j)$, we have

$$\frac{\sum_{j=1}^J u(a_j, q_j)}{\sum_{j=1}^J c(a_j, q_j)} = \frac{\sum_{a\in\mathcal{A}} N_a u(a, \bar{q}_a)}{\sum_{a\in\mathcal{A}} N_a c(a, \bar{q}_a)},$$

where

$$N_a = \sum_{j=1}^J \mathbb{I}\{a_j = a\}, \quad \bar{q}_a = \frac{1}{N_a}\sum_{j=1}^J \mathbb{I}\{a_j = a\}\, q_j\,.$$

Note that $\bar{q}_a \in \mathrm{conv}(Q_a)$ as it is a convex combination of $q_j \in Q_a$. Therefore, the definition in (19) is equivalent to

$$\rho_1(q) = \min\left\{ \frac{\sum_{a\in\mathcal{A}} \alpha_a u(a, q_a)}{\sum_{a\in\mathcal{A}} \alpha_a c(a, q_a)} \,:\, \alpha \in \Delta(\mathcal{A}), q_a \in \mathrm{conv}(Q_a), \sum_{a\in\mathcal{A}} \alpha_a q_a = q \right\}. \tag{20}$$

Now, this is exactly the definition of the so-called *calibration envelope* in Mannor and Shimkin (2008), and the claims of the lemma follow by Lemma 6.1 and Proposition 6.4 there. ■

It may be seen that $\rho_1(q)$ does not fall below the security level $\mathrm{val}(q)$, and is strictly above it when q is not a minimax action with respect to $\rho(p, q)$. Furthermore, at the vertices vertices of $\Delta(\mathcal{B})$, it actually coincides with the best ratio-in-hindsight $\rho^*(q)$.

We proceed to the following result that proves the attainability of $\rho_1(q)$.

Proposition 13 *Consider Algorithm 2 applied to the present model, with the goal function V^* defined in* (18)*. Thus, the agent's response q_n^* is chosen as any action $p_n^* \in P^*(q_n^*)$, and the target point is set to $v_n^* = (u(p_n^*, q_n^*), c(p_n^*, q_n^*))$. Then,*

$$\liminf_{n\to\infty} \left(\frac{\bar{U}_n}{\bar{C}_n} - \rho_1(\bar{q}_n) \right) \geq 0 \quad (a.s.)$$

for any strategy of the opponent.

Proof Algorithm 2 guarantees that, with probability 1,

$$\|\bar{q}_n - \bar{q}_n^*\| \to 0, \tag{21}$$

$$\left| \bar{U}_n - \frac{1}{n}\sum_{k=1}^n u(p_k^*, q_k^*) \right| \to 0, \quad \left| \bar{C}_n - \frac{1}{n}\sum_{k=1}^n c(p_k^*, q_k^*) \right| \to 0; \tag{22}$$

see Theorem 4 and recall the asymptotic equivalence of expected and actual averages. Noting that the cost c is positive and bounded away from zero, (22) implies that

$$\lim_{n\to\infty} \left| \frac{\bar{U}_n}{\bar{C}_n} - \frac{\sum_{k=1}^n r(p_k^*, q_k^*)}{\sum_{k=1}^n c(p_k^*, q_k^*)} \right| = 0. \tag{23}$$

Let

$$\rho_2(q) \triangleq \inf \left\{ \frac{\sum_{j=1}^J u(p_j, q_j)}{\sum_{j=1}^J c(p_j, q_j)} \,:\, J \geq 1,\, q_j \in \Delta(\mathcal{B}),\, \frac{1}{J}\sum_{j=1}^J q_j = q,\, p_j \in P^*(q_j) \right\}. \tag{24}$$

Clearly,

$$\frac{\sum_{k=1}^n r(p_k^*, q_k^*)}{\sum_{k=1}^n c(p_k^*, q_k^*)} \geq \rho_2(\bar{q}_n^*). \tag{25}$$

Furthermore, we verify below that the infimum in (24) is obtained in pure actions $a_j \in A^*(q_j)$, implying that

$$\rho_2(q) = \rho_1(q). \tag{26}$$

Indeed, note that the inequality

$$\frac{\sum_{j=1}^J u(p_j, q_j)}{\sum_{j=1}^J c(p_j, q_j)} \leq K$$

is equivalent to

$$\sum_{j=1}^J u(p_j, q_j) - K \sum_{j=1}^J c(p_j, q_j) \leq 0.$$

Now, consider minimizing the left-hand-side over $p_j \in P^*(q_j)$. Due to the linearity in p_j and the fact that $P^*(q_j)$ is just the mixture of actions in $A^*(q_j)$, the optimal actions are pure (that is, in $A^*(q_j)$).

Combining (23), (25), and (26), we obtain that

$$\liminf_{n\to\infty} \left(\frac{\bar{U}_n}{\bar{C}_n} - \rho_1(\bar{q}_n^*) \right) \geq 0.$$

The proof is concluded by applying (21) and the continuity (hence, uniform continuity) of ρ_1 (see Lemma 12). ■

We finally note that the algorithm variant from Subsection 4.3 can be applied here as well. Specifically, observe that the goal function V^* in (18) is unbounded in the u coordinate, and negatively unbounded in the c coordinate. Therefore, the u-coordinate of λ_n can be set to zero whenever $\bar{u}_{n-1} \geq \bar{u}_{n-1}^*$, while the c-coordinate of λ_n may be nullified whenever $\bar{c}_{n-1} \leq \bar{c}_{n-1}^*$.

5.2.3 Global Cost Functions

The following problem of regret minimization with global cost functions was introduced in Even-Dar et al. (2009). (A similar problem was recently addressed in Azar et al. (2014), using a relaxed regret criterion over sub-intervals.) Suppose that the goal of the agent is to minimize a general (i.e., non-linear) function of the average reward vector $\bar{v}_n$. In particular, we are given a *continuous* function $G : \mathbb{R}^K \to \mathbb{R}$, and the goal is to minimize $G(\bar{v}_n)$. For

example, G may be some *norm* of $\bar{v}_n$. We define the best-cost-in-hindsight, given a mixed action q of the opponent, as

$$G^*(q) \triangleq \min_{p \in \Delta(\mathcal{A})} G(v(p,q)), \tag{27}$$

so that the target set may be defined as

$$V^*(q) = \{v \in \mathcal{V}_0 : G(v) \le G^*(q)\}, \tag{28}$$

where $\mathcal{V}_0 = \mathrm{conv}\{v(a,b) : a \in \mathcal{A}, b \in \mathcal{B}\} \subset \mathbb{R}^K$ is the set of feasible reward vectors. Clearly, the agent's response to q is any mixed action that minimizes $G(v(p,q))$, namely

$$p^*(q) \in \operatorname*{argmin}_{p \in \Delta(\mathcal{A})} G(v(p,q)). \tag{29}$$

By Proposition 10, the closed convex hull $V^c = \overline{\mathrm{co}}(V^*)$ is attainable by the agent, and Algorithm 2 can be used to attain it. Observe that, in addition to solving a zero-sum matrix game, the algorithm requires solving the optimization problem (29). The computational complexity of the latter depends on the cost function G. For example, if G is convex, then (29) is a convex optimization problem. For specific instances, see Even-Dar et al. (2009) and Example 1 below.

The relation between V^c and V^* depends on the convexity properties of G and G^*. In particular, we have the following result (a slight extension of Even-Dar et al. (2009)).

Proposition 14 *For $q \in \Delta(\mathcal{B})$,*

$$V^c(q) \subset \tilde{V}(q) \triangleq \{v \in \mathcal{V}_0 \,:\, \mathrm{conv}(G)(v) \le \mathrm{conc}(G^*)(q)\}, \tag{30}$$

where $\mathrm{conv}(G)$ is the lower convex hull of G, and $\mathrm{conc}(G^*)$ is the upper concave hull of G^*. Consequently, any no-regret strategy with respect to $V^c = \overline{\mathrm{co}}(V^*)$ guaranties that, for any strategy of the opponent,

$$\limsup_{n\to\infty} \left(\mathrm{conv}(G)(\bar{v}_n) - \mathrm{conc}(G^*)(\bar{q}_n)\right) \le 0 \quad \textit{(a. s.)}. \tag{31}$$

In particular, if G is a convex function G^* a concave function, then $V^c = V^*$ and V^* itself is attained, namely

$$\limsup_{n\to\infty} \left(G(\bar{v}_n) - G^*(\bar{q}_n)\right) \le 0 \quad \textit{(a. s.)}.$$

Proof To show (30), recall that the graph of $V^c = \overline{\mathrm{co}}(V^*)$, by its definition, is given by

$$\mathrm{Graph}(V^c) = \overline{\mathrm{co}}(\mathrm{Graph}(V^*)),$$

and, by (28),

$$\mathrm{Graph}(V^*) = \{(v,q) \in \mathcal{V}_0 \times \Delta(\mathcal{B}) \,:\, G(v) \le G^*(q)\}.$$

Also, for $\tilde{V}$ as defined in (30),

$$\mathrm{Graph}(\tilde{V}) = \{(v,q) \in \mathcal{V}_0 \times \Delta(\mathcal{B}) \,:\, \mathrm{conv}(G)(v) \le \mathrm{conc}(G^*)(q)\}.$$

It is clear from these expressions that $\mathrm{Graph}(\tilde{V})$ is a convex set that contains $\mathrm{Graph}(V^*)$, hence $\mathrm{conv}(\mathrm{Graph}(V^*)) \subset \mathrm{Graph}(\tilde{V})$. Furthermore, since G is a continuous function by assumption, the G and G^* are continuous functions on compact sets, so that $\mathrm{conv}(G)$ and $\mathrm{conc}(G^*)$ are continuous functions, which implies that $\mathrm{Graph}(\tilde{V})$ is a closed set. Therefore $\overline{\mathrm{co}}(\mathrm{Graph}(V^*)) \subset \mathrm{Graph}(\tilde{V})$, and (30) follows. The other claims in the Proposition now follow directly from Proposition 10. ■

Clearly, if G^* is not concave, the attainable goal function is weaker than the original one. Still, this relaxed goal is meaningful, at least when G is convex. Noting the definition of G^* in (27), if follows that $G^*(q) \leq \max_{q'} \min_p G(v(p,q'))$ for all q, so that

$$\mathrm{conc}(G^*)(q) \leq \max_{q' \in \Delta(\mathcal{B})} \min_{p \in \Delta(\mathcal{A})} G(v(p,q')) \leq \min_{p \in \Delta(\mathcal{A})} \max_{q' \in \Delta(\mathcal{B})} G(v(p,q'))\,. \tag{32}$$

The latter min-max bound is just the security level of the agent in the repeated game, namely the minimal value of $G(\bar{v}_n)$ that can be secured (as $n \to \infty$) by playing a *fixed* (non-adaptive) mixed action q'. Note that the second inequality in Equation (32) will be strict except for special cases where the min-max theorem holds for $G(v(p,q))$ (which is hardly expected if $G^*(q)$ is non-concave).

Convexity of $G(v)$ depends on its definition, and will hold for cases of interest such as norm functions. Concavity of $G^*(q)$, on the other hand, is more demanding and will hold only in special cases. In Section 5.2.2 we already considered a specific instance of this model where $G(v) = -u/c$ is not convex and $G^*(q) = -\max_p\{u(p,q)/c(p,q)\}$ is not concave, hence specific analysis was required to obtain meaningful bounds. Another concrete model was considered in Even-Dar et al. (2009), motivated by load balancing and job scheduling problems. Under appropriate conditions, it was shown there that G is convex, while G^* can be seen to be concave, and the agent's response was computed in closed form. The details can be found in that reference and will not be elaborated here. These properties allow an easy application of Algorithm 2 above to attain V^* itself.

We close this section with a simple example, in which G is convex while G^* is not necessarily concave.

Example 1 (Absolute Value) Let $v : \mathcal{A} \times \mathcal{B} \to \mathbb{R}$ be a scalar reward function, and suppose that we wish to minimize the deviation of the average reward $\bar{v}_n$ from a certain preset value, say 0. Define then $G(v) = |v|$, and note that G is a convex function. Now,

$$G^*(q) \triangleq \min_{p \in \Delta(\mathcal{A})} |v(p,q)| = \begin{cases} \min_{a \in \mathcal{A}} v(a,q), & \text{if } \forall a \in \mathcal{A},\ v(a,q) > 0 \\ \min_{a \in \mathcal{A}} (-v(a,q)), & \text{if } \forall a \in \mathcal{A},\ v(a,q) < 0 \\ 0, & \text{otherwise.} \end{cases}$$

The response $p^*(q)$ of the agent is obvious from these relations. We can observe two special cases in this example:

(*i*) The problem reduces to the classical no-regret problem if the rewards $v(a,b)$ all have the same sign (positive or negative), as the absolute value can be removed. Indeed, in this case $G^*(q)$ is concave, as a minimum of linear functions.

(*ii*) If the set $\{v(a,q), a \in \mathcal{A}\}$ includes elements of opposite signs (0 included) for each q, then $G^* = 0$, and the point $v = 0$ becomes attainable.

In general, however, $|v(p,q)|$ may be a *strictly* convex function of q for a fixed p, and the minimization above need not lead to a concave function. In that case, Proposition 14 implies only the attainability of $\mathrm{conc}(G^*)(q)$.

We note that the computation of $\mathrm{conc}(G^*)$ may be fairly complicated in general, which implies the same for computing the projection onto the associated goal set $S = \{(v,q) : |v| \le \mathrm{conc}(G^*)(q)\}$. However, these computations are not needed in the response-based approachability algorithm, where the required computation of the agent's response $p^*(q)$ is straightforward.

6. Conclusion

We have introduced in this paper an approachability algorithm that is based on Blackwell's dual, rather than primal, approachability condition. The proposed algorithm and its variants rely directly on the availability of a response function, rather than projection onto the goal set (or related geometric quantities), and are therefore convenient in problems where the latter may be hard to compute. At the same time, the additional computational requirements are generally comparable to those of the standard Blackwell algorithm and its variants.

The proposed algorithms were applied to a class of generalized no-regret problems, that includes as specific cases the constrained no-regret problem and reward-to-cost maximization. The resulting algorithms are apparently the first computationally efficient algorithms in this generalized setting.

In this paper we have focused on a repeated *matrix* game model, where the action sets of the agent and the opponent in the stage game are both finite. It is worth pointing out that the essential results of this paper should apply directly to models with *convex* action sets, say X and Y, provided that the reward vector $r(x,y)$ is bilinear in its arguments. In that case the (observed) actions x and y simply take the place of the mixed actions p and q, leading to similar algorithms and convergence results. Such a continuous-action model is relevant to linear classification and regression problems.

Other extensions of possible interest for the approach of this paper may include stochastic game models, problems of partial monitoring, and nonlinear (concave-convex) reward functions. These are left for future work.

Acknowledgements

Most of this work was done while the first author was a PhD student at the Department of Electrical Engineering, Technion. This research was supported by the Israel Science Foundation grant No. 1319/11. We wish to thank Shie Mannor for useful discussions, and for pointing out the application to regret minimization with global cost functions. We also thank two anonymous reviewers for their useful comments on this manuscript that helped improve the presentation as well as certain technical aspects.

References

J. Abernethy, P. L. Bartlett, and E. Hazan. Blackwell approachability and low-regret learning are equivalent. In *Proceedings of the 24th Conference on Learning Theory (COLT'11)*, pages 27–46, Budapest, Hungary, June 2011.

J.-P. Aubin and H. Frankowska. *Set-Valued Analysis*. Birkhauser, Boston, MA, 1990.

R.J. Aumann and M. Maschler. *Repeated Games with Incomplete Information*. MIT Press, Boston, MA, 1995.

Y. Azar, U. Felge, M. Feldman, and M. Tennenholtz. Sequential decision making with vector outcomes. In *Proceedings of the 5th Conference on Innovations in Theoretical Computer Science (ITCS'14)*, pages 195–206, January 2014.

A. Bernstein. *Approachability in Dynamic Games: Algorithms, Refinements, and Applications to No-Regret Problems*. PhD thesis, Technion, Haifa, Israel, October 2013.

A. Bernstein and N. Shimkin. Response-based approachability with applications to generalized no-regret problems. October 2013. Preprint, http://arxiv.org/abs/1312.7658.

A. Bernstein, S. Mannor, and N. Shimkin. Online classification with specificity constraints. In *Proceedings of the 23rd Conference on Neural Information Processing Systems (NIPS'10)*, pages 190–198, Vancouver, Canada, December 2010.

A. Bernstein, S. Mannor, and N. Shimkin. Opportunistic approachability and generalized no-regret problems. *Mathematics of Operations Research*, 39(4):1057–1083, 2014. Also in *Proc. COLT 2013*.

D. Blackwell. Controlled random walks. In *Proceedings of the International Congress of Mathematicians*, volume III, pages 335–338, 1954.

D. Blackwell. An analog of the minimax theorem for vector payoffs. *Pacific Journal of Mathematics*, 6:1–8, 1956.

N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, New York, NY, 2006.

E. Even-Dar, R. Kleinberg, S. Mannor, and Y. Mansour. Online learning with global cost functions. In *Proceedings of the 22nd Conference on Learning Theory (COLT'09)*, 2009.

D. Foster. A proof of calibration via Blackwell's approachability theorem. *Games and Economic Behavior*, 29:73–78, 1999.

D. Fudenberg and D. K. Levine. *The Theory of Learning in Games*. MIT Press, Boston, MA, 1998.

J. Hannan. Approximation to Bayes risk in repeated play. *Contributions to the Theory of Games*, 3:97–139, 1957.

S. Hart and A. Mas-Colell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68:1127–1150, 2000.

S. Hart and A. Mas-Colell. A general class of adaptive strategies. *Journal of Economic Theory*, 98:26–54, 2001.

E. Hazan and S. Kakade. (weak) Calibration is computationally hard. In *Proceeding of the 25th Conference on Learning Theory (COLT'12)*, pages 3.1–3.10, Edinburgh, Scotland, June 2012.

E. Lehrer. Approachability in infinite dimensional spaces. *International Journal of Game Theory*, 31:253–268, 2002.

E. Lehrer and E. Solan. Learning to play partially-specified equilibrium. Manuscript, available online: `http://www.math.tau.ac.il/~lehrer/Papers/LearningPSCE-web.pdf`, 2007.

E. Lehrer and E. Solan. Approachability with bounded memory. *Games and Economic Behavior*, 66(2):995–1004, 2009.

P. Maċkowiak. Some remarks on lower hemicontinuity of convex multivalued mappings. *Economic Theory*, 28(1):227–233, 2006.

S. Mannor and N. Shimkin. The empirical Bayes envelope and regret minimization in competitive Markov decision processes. *Mathematics of Operations Research*, 28(2):327–345, 2003.

S. Mannor and N. Shimkin. A geometric approach to multi-criterion reinforcement learning. *Journal of Machine Learning Research*, 5:325–360, 2004.

S. Mannor and N. Shimkin. Regret minimization in repeated matrix games with variable stage duration. *Games and Economic Behavior*, 63(1):227–258, 2008.

S. Mannor, J. N. Tsitsiklis, and J. Y. Yu. Online learning with sample path constraints. *Journal of Machine Learning Research*, 10:569–590, 2009.

S. Mannor, V. Perchet, and G. Stoltz. Approachability in unknown games: Online learning meets multi-objective optimization. In *Proceeding of the 27th Conference on Learning Theory (COLT'14)*, pages 339–355, Barcelona, Spain, May 2014.

J.F. Mertens, S. Sorin, and S. Zamir. *Repeated Games*. CORE Discussion Papers 9420-9422, Universitè Catholique de Louvain, 1994.

V. Perchet. Calibration and internal no-regret with partial monitoring. In *Proceedings of the 20th International Conference on Algorithmic Learning Theory (ALT '09)*, Porto, Portugal, October 2009.

V. Perchet. Approachability, regret and calibration: Implications and equivalences. *Journal of Dynamics and Games*, 1:181–254, 2014.

R.T. Rockafellar and R. Wets. *Variational Analysis*. Springer-Verlag, 1997.

A. Rustichini. Minimizing regret: The general case. *Games and Economic Behavior*, 29: 224–243, 1999.

N. Shimkin and A. Shwartz. Guaranteed performance regions in Markovian systems with competing decision makers. *IEEE Transactions on Automatic Control*, 38(1):84–95, 1993.

X. Spinat. A necessary and sufficient condition for approachability. *Mathematics of Operations Research*, 27(1):31–44, 2002.

N. Vieille. Weak approachability. *Mathematics of Operations Research*, 17(4):781–791, 1992.

H. P. Young. *Strategic Learning and Its Limits*. Oxford University Press, 2004.

Journal of Machine Learning Research 16 (2015) 775-785 Submitted 12/13; Revised 10/14; Published 4/15

Strong Consistency of the Prototype Based Clustering in Probabilistic Space

Vladimir Nikulin VNIKULIN.UQ@GMAIL.COM
Department of Mathematical Methods in Economy
Vyatka State University, Kirov, 610000, Russia

Editor: Inderjit Dhillon

Abstract

In this paper we formulate in general terms an approach to prove strong consistency of the Empirical Risk Minimisation inductive principle applied to the prototype or distance based clustering. This approach was motivated by the Divisive Information-Theoretic Feature Clustering model in probabilistic space with Kullback-Leibler divergence, which may be regarded as a special case within the Clustering Minimisation framework.

Keywords: clustering, probabilistic space, consistency

1. Introduction

Clustering algorithms group objects into subsets (clusters) of similar items according to the given criteria. For example, it may be Spectral Clustering (Ng et al., 2001) or Prototype Based model (Hinneburg and Keim, 2003). Clustering has application in various areas of computer science such as machine learning, data compression, data mining or patterns recognition. Depending on the area of application, there are many different formulations of the clustering problem (Ackerman et al., 2008). For example, we can consider text document as an object with words as features, and the task is to cluster text documents into subsets, corresponding to a few given topics. This problem maybe effectively approximated by the clustering model in probabilistic space with Kullback Leibler (KL) divergence (Dhillon et al., 2003) which arises as a natural measure of the dissimilarity between two distributions in numerical way. Further related results are presented by Chaudhuri and McGregor (2008), where authors provide algorithms for clustering using the KL-divergence measure with an objective to achieve guaranteed approximation in the worst case.

In this paper we consider a prototype based approach which may be described as follows. Initially, we have to choose k prototypes. The corresponding empirical clusters will be defined in accordance to the criteria of the nearest prototype measured by the distance Φ. Respectively, we will generate initial k clusters. As a second Minimisation step, we shall recompute cluster centers or Φ-means (Cuesta-Albertos et al., 1997), using data strictly from the corresponding clusters. Then, we can repeat Clustering step, using new prototypes, obtained from the previous step as cluster centers. Above algorithm has descending property. Respectively, it will reach local minimum in a finite number of steps.

Stability is a common tool to verify the validity of sample based algorithms. Clustering is one of the most widely used techniques for exploratory data analysis. Across all disciplines, from social sciences over biology to computer science, people try to get a first intuition about

their data by identifying meaningful groups among the data points. Despite this popularity of clustering, distressingly little is known about theoretical properties of clustering (Ben-David et al., 2006).

1.1 Related Work

One formulation of stability is: if parameters are learned over two different samples from the same distribution, how close they are? The statistical stability for clustering have been extensively studied (Rakhlin and Caponnetto, 2006; Shamir and Tishby, 2008).

Pollard (1981) demonstrated that the classical K-means algorithm in $\mathbb{R}^m$ with squared loss function satisfies the Key Theorem of Learning Theory (Vapnik, 1995), p.36, *"the minimal empirical risk must converge to the minimal actual risk"*. Note, also study (Biau et al., 2008), where the number of theorems that establish the universal consistency of averaging rules are given.

Telgarsky and Dasgupta (2013) constructed an explicit moment-based uniform deviation bounds for the convergence of the soft clustering processes in Euclidean space. The results of Telgarsky and Dasgupta (2013) are general and significant: assuming that some probability moments are limited, an explicit convergence bounds are constructed. Compared to the model of Pollard (1981), the framework presented by Telgarsky and Dasgupta (2013) represents a novel direction, and we are considering to extend it to the probabilistic space in our future work.

Note that both functions 1) squared loss and 2) KL divergence are covered by the general structural definition of *Bregman* divergence. Bregman divergences give us a lot of freedom in fitting the performance measure of our algorithm to the nature of the data, and, as a consequence, this will lead to qualitatively better clustering (Banerjee et al., 2005), (Nock et al., 2008). Bregman divergences have found many applications in the fields of machine learning and computational geometry.

A new clustering algorithm in probabilistic space $\mathcal{P}^m$ was proposed by Dhillon et al. (2003). It provides an attractive approach based on the Kullback-Leibler divergence. The above methodology requires a general formulation and framework which we present in the following Section 2.

There are many useful and popular models and algorithms in the field of machine learning in addition to clustering. Consistency of those models represents a very essential property which should be investigated. For example, the subject of the papers (Glasmachers, 2010), (McAllester and Keshet, 2011) is consistency of support vector classifiers. Also, it is very interesting to identify those models, which are not consistent (Long and Servedio, 2013), and explain the reasons for not consistency. Particularly interesting is to find general conditions under which the common approaches, with various algorithmic variations, are consistent (Kpotufe et al., 2014).

1.2 Structure of the Paper

The paper is organised as follows. Section 3 extends the methodology of Pollard (1981) in order to cover the case of $\mathcal{P}^m$ with Kullback-Leibler divergence. With the aim to highlight the most essential properties, we formulate the model in general terms, where the probabilistic space is considered as an important example. We do believe that the structural

approach, which is formulated in Section 3, maybe useful for the consideration of other cases (different space or the same space with different loss function). In this sense our work is similar to Kpotufe et al. (2014). Using the results and definitions of the Section 3, we investigate relevant properties of $\mathcal{P}^m$ in the final Section 4 and prove a strong consistency of the Empirical Risk Minimisation inductive principle.

2. Prototype Based Approach

In this paper we consider a sample of i.i.d. observations $\mathbf{X} := \{x_1, \ldots, x_n\}$ drawn from probability space $(\mathcal{X}, \mathcal{A}, \mathbb{P})$ where probability measure $\mathbb{P}$ is assumed to be unknown.

Key in this scenario is an encoding problem. Assuming that we have a codebook $\mathcal{Q} \in \mathcal{X}^k$ with *prototypes* $q(c)$ indexed by the code $c = 1, \ldots, k$, the aim is to encode any $x \in \mathcal{X}$ by some $q(c(x))$ such that the distortion between x and $q(c(x))$ is minimized:

$$c(x) := \operatorname*{argmin}_c \mathcal{L}(x, q(c)), \tag{1}$$

where $\mathcal{L}(\cdot, \cdot)$ is a loss function.

Using criterion (1) we split empirical data into k clusters. As a next step we compute the *cluster center* specifically for any particular cluster in order to minimise overall distortion error.

We estimate actual distortion error

$$\Re^{(k)}[\mathcal{Q}] := \mathbf{E} \;\; \mathcal{L}(x, \mathcal{Q}) \tag{2}$$

by the empirical error

$$\Re^{(k)}_{\text{emp}}[\mathcal{Q}] := \frac{1}{n} \sum_{t=1}^{n} \mathcal{L}(x_t, \mathcal{Q}), \tag{3}$$

where $\mathcal{L}(x, \mathcal{Q}) := \mathcal{L}(x, q(c(x)))$.

The following Theorem, which may be proved similarly to the Theorems 4 and 5 (Dhillon et al., 2003), formulates the most important descending and convergence properties within the *Clustering Minimisation* (CM) framework:

Theorem 1 *The CM-algorithm includes 2 steps:*

***Clustering Step**: recompute $c(x)$ according to (1) for a fixed prototypes from the given codebook $\mathcal{Q}$, which will be updated as a cluster centers from the next step,*

***Minimisation Step**: recompute cluster centers for a fixed mapping $c(x)$ or minimize the objective function (3) over $\mathcal{Q}$, and*

1) monotonically decreases the value of the objective function (3);

2) converges to a local minimum in a finite number of steps if Minimisation Step has exact solution.

We define an optimal actual codebook $\overline{\mathcal{Q}}$ by the following condition:

$$\Re^{(k)}(\overline{\mathcal{Q}}) := \inf_{\mathcal{Q} \in \mathcal{X}^k} \Re^{(k)}(\mathcal{Q}). \tag{4}$$

The following relations are valid

$$\Re^{(k)}_{\text{emp}}[\mathcal{Q}_n] \le \Re^{(k)}_{\text{emp}}[\overline{\mathcal{Q}}]; \quad \Re^{(k)}_{\text{emp}}[\overline{\mathcal{Q}}] \Rightarrow \Re^{(k)}[\overline{\mathcal{Q}}] \quad a.s., \tag{5}$$

where $\mathcal{Q}_n$ is an optimal empirical codebook:

$$\Re^{(k)}_{\text{emp}}(\mathcal{Q}_n) := \inf_{\mathcal{Q}\in\mathcal{X}^k} \{\Re^{(k)}_{\text{emp}}(\mathcal{Q})\}. \tag{6}$$

The main target is to demonstrate asymptotic (*almost sure*) convergence

$$\Re^{(k)}_{\text{emp}}(\mathcal{Q}_n) \Rightarrow \Re^{(k)}[\overline{\mathcal{Q}}] \quad a.s. \quad (n \to \infty)\,. \tag{7}$$

In order to prove (7) we define in Section 3 general model which has direct relation to the model in probabilistic space $\mathcal{P}^m$ with with KL divergence (Dhillon et al., 2003).

2.1 Plan of the Proof

The general strategy is to split consideration into outer deviations, and local deviations (Telgarsky and Dasgupta, 2013). Note that the significance of outer deviations is declining as we extend local deviation. The local deviations maybe be controlled by the technique as described below.

The proof of the main result which is formulated in the Theorem 18 includes two steps:

(1) by Lemma 10 we prove existence of n_0 such that $\mathcal{Q}_n \subset \Gamma$ for all $n \ge n_0$, where subset $\Gamma \subset \mathcal{X}$ (local deviation) satisfies condition: $\mathcal{L}(x,q) < \infty$ for all $x \in \mathcal{X}, q \in \Gamma$; and

(2) by Lemma 11 we prove (under some additional constraints of general nature)

$$\sup_{\mathcal{Q}\in\Gamma^k} |\Re^{(k)}_{\text{emp}}[\mathcal{Q}] - \Re^{(k)}[\mathcal{Q}]| \Rightarrow 0 \quad a.s. \tag{8}$$

3. General Theory and Definitions

In this section we employ some ideas and methods proposed by Pollard (1981) which cover the case of $\mathbb{R}^m$ with loss function $\mathcal{L}(x,q) := \varphi(\|x-q\|)$, where φ is a strictly increasing function.

Let us assume that the following structural representation with $\mathbb{P}$-integrable vector-functions ξ and η is valid

$$\mathcal{L}(x,q) := \sum_{i=0}^{m} \xi_i(x)\cdot\eta_i(q) = \langle \xi(x), \eta(q)\rangle \ge 0 \quad \forall x,q \in \mathcal{X}. \tag{9}$$

Remark 2 *Above definition (9) was motivated by the structure of KL-divergence, see (29a) and (29b).*

Let us define subsets of $\mathcal{X}$ as extensions of the empirical clusters:

$$\mathcal{X}_c(\mathcal{Q}) := \{x \in \mathcal{X} : \quad c = \text{argmin}_i\, \mathcal{L}(x, q(i))\}\,,$$

$$\mathcal{X} = \cup_{c=1}^{k} \mathcal{X}_c(\mathcal{Q}), \mathcal{X}_i(\mathcal{Q}) \cap \mathcal{X}_c(\mathcal{Q}) = \emptyset, i \neq c.$$

Then, we re-write (2) as follows

$$\Re^{(k)}[\mathcal{Q}] := \sum_{c} \langle \xi(\mathcal{X}_c), \eta(q(c)) \rangle, \tag{10}$$

where $\xi(A) := \int_A \xi(x)\mathbb{P}(dx), A \in \mathcal{A}$.

Definition 3 *We define a ball with radius r and a corresponding remainder in $\mathcal{X}$*

$$B(r) = \{q \in \mathcal{X} : \mathcal{L}(x, q) \leq r, \quad \forall x \in \mathcal{X}\}, \tag{11a}$$

$$T(r) = \mathcal{X} \setminus B(r), \quad r \geq \mathbf{r}_0, \tag{11b}$$

$$\mathbf{r}_0 = \inf\{r \geq 0 : B(r) \neq \emptyset\}. \tag{11c}$$

Remark 4 *By the following Lemma 10 we prove that all components of the codebook will be within ball $B(Z), 0 < Z < \infty$, if sample size is large enough. Further, we shall assume that η-transformation of the ball $B(Z)$ represents a compact set (26), and, consequently, we shall be able to prove strong consistency (8) by Lemma 11.*

The following properties are valid

$$\langle \xi(A_1) - \xi(A_2), \eta(q) \rangle \geq 0 \tag{12}$$

for all $q \in \mathcal{X}$ and any $A_1, A_2 \in \mathcal{A} : A_2 \subset A_1$;

$$\langle \xi(\mathcal{X}), \eta(q) \rangle \leq r \quad \forall q \in B(r). \tag{13}$$

Suppose, that

$$\mathbb{P}(T(U)) \underset{U \to \infty}{\longrightarrow} 0. \tag{14}$$

Remark 5 *Condition (14) is the only one requirement which is necessary to prove the main result of this paper: Theorem 18, see, also, Remark 19.*

Definition 6 *The following distances will be used below:*

$$\rho(A_1, A_2) := \underset{a_1 \in A_1 a_2 \in A_2}{inf} \; \underset{}{inf} \; \mathcal{L}(a_1, a_2), A_1, A_2 \in \mathcal{A}; \tag{15a}$$

$$\mu(A_1, A_2) := \underset{a_1 \in A_1 a_2 \in A_2}{inf} \; \underset{}{sup} \; \mathcal{L}(a_1, a_2), A_1, A_2 \in \mathcal{A}. \tag{15b}$$

Remark 7 *Above distances ρ and μ have very simple interpretation: ρ - absolutely minimal distance between elements of the subsets A_1 and A_2 (it is symmetrical); μ - uniformly minimal distance between elements of the subset A_1 (approximator) and elements of another subset A_2 (it is not symmetrical).*

Suppose, that

$$\rho(B(r), T(U)) \underset{U\to\infty}{\longrightarrow} \infty \tag{16}$$

for any fixed $\mathbf{r}_0 \le r < \infty$.

Remark 8 *Above condition (16) is always valid for KL-divergence, see Corollary 17.*

Remark 9 *We assume that*

$$T(U) \neq \emptyset \tag{17}$$

for any fixed $U : \mathbf{r}_0 \le U < \infty$, alternatively, the following below Lemma 10 becomes trivial.

Lemma 10 *Suppose, that the structure of the loss function $\mathcal{L}$ is defined in (9) under condition (16). Probability distribution $\mathbb{P}$ satisfies condition (14) and the number of clusters $k \ge 1$ is fixed. Then, we can select large enough radius $Z : 0 < Z < \infty$ and $n_0 \ge 1$ such that all components of the optimal empirical codebook $\mathcal{Q}_n$ defined in (6) will be within the ball $B(Z)$: $\mathcal{Q}_n \subset B(Z)$ if sample size is large enough: $\forall n \ge n_0$.*

Proof. Existence of the element $\mathbf{a} \in \mathcal{X}$ such that

$$D_{\mathbf{a}} = \Re^{(1)}(\{\mathbf{a}\}) = \langle \xi(\mathcal{X}), \eta(\mathbf{a}) \rangle < \infty \tag{18}$$

follows from (13) and (14).

Suppose that

$$\mathbb{P}(B(r)) = P_0 > 0, \;\; r \ge \mathbf{r}_0. \tag{19}$$

We construct $B(V)$ in accordance with conditions (16) and (17):

$$V = \inf\{v > r : \rho(B(r), T(v)) \ge \frac{D_{\mathbf{a}} + \epsilon}{P_0}\}, \;\; \epsilon > 0. \tag{20}$$

Suppose, there are no empirical prototypes within $B(V)$. Then, in accordance with definition (19)

$$\Re^{(k)}_{\text{emp}}[\mathcal{Q}_n] \ge D_{\mathbf{a}} + \epsilon > D_{\mathbf{a}} \;\; \forall n > 0.$$

Above contradicts to (18) and (5). Therefore, at least one prototype from $\mathcal{Q}_n$ must be within $B(V)$ if n is large enough (this fact is valid for $\overline{\mathcal{Q}}$ as well). Without loss of generality we assume that

$$q(1) \in B(V). \tag{21}$$

The proof of the Lemma has been completed in the case if $k = 1$.

Assumption. *Following the method of mathematical induction, suppose, that $k \ge 2$ and*

$$\Re^{(k-1)}(\overline{\mathcal{Q}}) - \Re^{(k)}(\overline{\mathcal{Q}}) \ge \varepsilon > 0. \tag{22}$$

Then, we define a ball $B(U)$ by the following conditions

$$U = \inf\{u > V : \sup_{q \in B(V)} \langle \xi(T(u)), \eta(q) \rangle < \varepsilon\}. \tag{23}$$

Existence of the $U : V < U < \infty$ in (23) follows from (13) and (14).

By definition of the distance μ and the ball $B(V)$

$$0 < \mathcal{D}(U, V) = \mu(T(U), B(V)) \leq V < \infty. \tag{24}$$

Now, we define reminder $T(Z) \neq \emptyset$ in accordance with condition (16):

$$Z = \inf\{z > U : \rho(B(U), T(z)) \geq \mathcal{D}(U, V)\}. \tag{25}$$

Suppose, that there is at least one prototype within $T(Z)$, for example, $q(2) \in T(Z)$. On the other hand, we know about (21). Let us consider what will happen if we remove $q(2)$ from the optimal empirical codebook $\mathcal{Q}_n$ (the case of optimal actual risk $\overline{\mathcal{Q}}$ may be considered similarly), and replace it by $q(1)$:

(1) as a consequence of (24) and (25) all empirical data within $B(U)$ are closer to $q(1)$ anyway, means the data from $B(U)$ will not increase empirical (or actual) risk (3);

(2) by definition, $\mathcal{X} = B(U) \cup T(U), B(U) \cap T(U) = \emptyset$ and in accordance with the condition (23) an empirical risk increases because of the data within $T(U)$ must be strictly less compared with ε for all large enough $n \geq n_0$ (actual risk increase will be strictly less compared with ε for all $n \geq 1$).

Above *contradicts* to the condition (22) and (5). Therefore, all prototypes from $\overline{\mathcal{Q}}$ must be within $\Gamma = B(Z)$ for all $n \geq 1$, and $\mathcal{Q}_n \subset \Gamma$ if n is large enough. ■

3.1 Uniform Strong Law of Large Numbers (SLLN)

Let $\mathcal{F}$ denote the family of $\mathbb{P}$-integrable functions on $\mathcal{X}$.

A sufficient condition for *uniform SLLN* (8) is: for each $\delta > 0$ there exists a *finite* class $\mathcal{F}_\delta \in \mathcal{F}$ such that for each $\mathcal{L} \in \mathcal{F}$ there are functions $\underline{\mathcal{L}}$ and $\overline{\mathcal{L}} \in \mathcal{F}_\delta$ with the following 2 properties:

$$\underline{\mathcal{L}}(x) \leq \mathcal{L}(x) \leq \overline{\mathcal{L}}(x) \text{ for all } x \in \mathcal{X}; \quad \int_{\mathcal{X}} \left(\overline{\mathcal{L}}(x) - \underline{\mathcal{L}}(x)\right) \mathbb{P}(dx) \leq \delta.$$

We assume here existence of the function φ such that

$$\|\eta(q)\| \leq \varphi(Z) < \infty \tag{26}$$

for all $q \in B(Z)$, where $\mathbf{r}_0 \leq Z < \infty$.

Lemma 11 *Suppose that the number of clusters k is fixed, and the loss function $\mathcal{L}$ is defined by (9) under condition (26) and*

$$\|\xi(x)\| \leq \mathbf{R} < \infty \ \ \forall x \in \mathcal{X}. \tag{27}$$

Then, the asymptotic relation (8) is valid for any $\Gamma = B(Z), \mathbf{r}_0 \leq Z \leq \infty$.

Proof. Let us consider the definition of Hausdorff metric $\mathcal{H}$ in $\mathbb{R}^{m+1}$:

$$\mathcal{H}(A_1, A_2) = \underset{a_1 \in A_1}{sup} \underset{a_2 \in A_2}{inf} \|a_1 - a_2\|,$$

and denote by $\mathcal{G}$ a subset in $\mathbb{R}^{m+1}$ which was obtained from Γ as a result of η-transformation. According to the condition (26), $\mathcal{G}$ represents a compact set. It means, existence of a finite subset $\mathcal{G}_\delta$ for any $\delta > 0$ such that $\mathcal{H}(\mathcal{G}, \mathcal{G}_\delta) \leq \frac{\delta}{2\mathbf{R}}$, where $\mathbf{R}$ is defined in (27). We denote by $\Gamma_\delta \subset \Gamma$ subset which corresponds to $\mathcal{G}_\delta \subset \mathcal{G}$ according to the η-transformation. Respectively, we can define transformation (according to the principle of the nearest point) f_δ from Γ to Γ_δ, and $\mathcal{Q}_\delta = f_\delta(\mathcal{Q})$, where closeness may be tested independently for any particular component of $\mathcal{Q}$, that means absolute closeness.

In accordance with the Cauchy-Schwartz inequality, the following relations take place

$$\underline{\mathcal{L}} = \mathcal{L}(x, \mathcal{Q}_\delta) - \frac{\delta}{2} \leq \mathcal{L}(x, \mathcal{Q}) \leq \mathcal{L}(x, \mathcal{Q}_\delta) + \frac{\delta}{2} = \overline{\mathcal{L}} \; \forall x \in \mathcal{X}.$$

Finally, $\int_{\mathcal{X}} \left(\overline{\mathcal{L}}(x, \mathcal{Q}_\delta) - \underline{\mathcal{L}}(x, \mathcal{Q}_\delta)\right) \mathbb{P}(dx) \leq \delta$, where $\mathcal{Q}_\delta \in \Gamma_\delta^k$ is the absolutely closest codebook for the arbitrary $\mathcal{Q} \in \Gamma^k$. ■

4. A Probabilistic Framework

Following Dhillon et al. (2003), we assume that the probabilities $p_{\ell t} = P(\ell | x_t), \sum_{\ell=1}^m p_{\ell t} = 1, t = 1, \ldots, n$, represent relations between observations x_t and attributes or classes $\ell = 1, \ldots, m, m \geq 2$.

Accordingly, we define probabilistic space $\mathcal{P}^m$ of all m-dimensional probability vectors with *Kullback-Leibler* (KL) divergence:

$$KL(v, u) := \sum_\ell v_\ell \cdot \log \frac{v_\ell}{u_\ell} = \langle v, \log \frac{v}{u} \rangle, \; v, u \in \mathcal{P}^m.$$

Remark 12 *As it was demonstrated by Dhillon et al. (2003), cluster centers q_c in the space $\mathcal{P}^m$ with KL-divergence must be computed using K-means:*

$$q_c = \frac{1}{n_c} \sum_{x_t \in \mathbf{X}_c} p_t, \tag{28}$$

where $c(x_t) = c$ if $x_t \in \mathbf{X}_c$ and $n_c = \#\mathbf{X}_c$ is the number of observations in the cluster $\mathbf{X}_c, c = 1, \ldots, k$, $p_t = \{p_{1t}, \ldots, p_{mt}\}, q_c = \{q_{1t}, \ldots, q_{mt}\}$.

In difference to the model of Pollard (1981) in $\mathbb{R}^m$, the structure (9) covers an important case of $\mathcal{P}^m$ with KL-divergence:

$$\xi_0(v) = \sum_{\ell=1}^m v_\ell \log v_\ell; \quad \xi_\ell(v) = v_\ell; \tag{29a}$$

$$\eta_0(u) = 1; \quad \eta_\ell(u) = -\log u_\ell, \ell = 1, \ldots, m. \tag{29b}$$

Definition 13 *We call element $v \in \mathcal{P}^m$ as 1) uniform center if $v_\ell = \frac{1}{m}, \ell = 1, \ldots, m$; as 2) absolute margin if $\min_\ell v_\ell = 0$.*

Proposition 14 *The ball $B(Z) \subset \mathcal{P}^m$ contains only one element named as uniform center in the case if $Z = \mathbf{r}_0 = \log(m)$, and $B(Z) = \emptyset$ if $Z < \mathbf{r}_0$.*

Proof. Suppose, that u is a uniform center. Then, $KL(v,u) = \sum_{i=1}^m v_i \log v_i + \log m \leq \log m$ for all $v \in \mathcal{P}^m$. In any other case, one of the components of u must be less than $\frac{1}{m}$. Respectively, we can select the corresponding component of probability vector v as 1. Therefore, $KL(v,u) > \log(m)$ and $\mathbf{r}_0 = \log(m)$. ■

Lemma 15 *The KL divergence in probabilistic space $\mathcal{P}^m$ always satisfies condition (27), where vector-function ξ is expressed by (29a)with the following upper bounds:*

$$|\xi_0(v)| \leq \log(m); \quad |\xi_\ell(v)| \leq 1, \ell = 1, \ldots, m, \qquad \forall v \in \mathcal{P}^m.$$

Lemma 16 *The following relations are valid in $\mathcal{P}^m$*

(1) $min_\ell\{u_\ell\} < e^{-r}$ for all $u \in T(r) \;\; \forall r \geq \mathbf{r}_0$;

(2) $u_\ell \geq e^{-r}$ for all $\ell = 1, \ldots, m$, and any $u \in B(r) \;\; \forall r \geq \mathbf{r}_0$.

Proof. As far as $\mathcal{P}^m = B(r) \cup T(r), B(r) \cap T(r) = \emptyset$, the first statement may be regarded as a consequence of the second statement. Suppose, that $u \in B(r)$ and $u_1 = e^{-r-\varepsilon}, \varepsilon > 0$. Then, we can select $v_1 = 1$, and $KL(v,u) = r + \varepsilon > r$ - *contradiction.* ■

Corollary 17 *The KL divergence in $\mathcal{P}^m$ always satisfies conditions (16), and*

$$-\log(m) + Z \cdot e^{-r} < \rho(B(r), T(Z)) \leq e^{-r} \cdot (Z - r) + \left(1 - e^{-r}\right) \log \frac{1 - e^{-r}}{1 - e^{-Z}}$$

for all $\mathbf{r}_0 \leq r < Z$, where the distance ρ is defined in (15a).

Proof. Suppose, that $v \in B(r)$ and $u \in T(Z)$. Then, $-\sum_{i=1}^m v_i \log(u_i) > Z \cdot e^{-r}$ for all $r : \mathbf{r}_0 \leq r < Z$. On the other hand, the entropy $H(v) = -\sum_{i=1}^m v_i \log(v_i)$ may not be smaller compared to $\log(m)$. The low bound is *proved*. In order to prove the upper bound we suppose without loss of generality that $v_1 = e^{-r}, u_1 = e^{-Z}$, and all the other components are proportional. ■

Theorem 18 *Suppose that probability measure $\mathbb{P}$ satisfies condition (14) in probabilistic space $\mathcal{P}^m$ with KL divergence and the number of clusters k is fixed. Then, the minimal empirical error (6) converges to the minimal actual error (4) with probability 1 or a.s.*

Proof. Follows directly from the Lemmas 10, 11, 15 and 16.

Remark 19 *Condition (14) is not valid if and only if the probability of the subset of all absolute margins is strictly positive. Note that in order to avoid any problems with consistency we can generalise definition of KL-divergence using special smoothing parameter $0 \leq \theta \leq 1$:*

$$KL_\theta(v,u) = KL(v_\theta, u_\theta),$$

where $v_\theta = \theta v + (1-\theta)v_0$, and $u_\theta = \theta u + (1-\theta)v_0$, v_0 is uniform center.

5. Concluding Remarks

Consistency is a key property of all statistical procedures analyzing randomly sampled data. Surprisingly, despite decades of work, little is known about consistency of most clustering algorithms (von Luxburg et al., 2008). In this paper we developed a general framework to investigate and to prove consistency of the popular family of prototype based clustering algorithms. As an illustration, we considered probabilistic space with Kullback-Leibler divergence.

Acknowledgment

The author would like to thank two reviewers for very helpful comments and advice.

References

M. Ackerman, J. Blomer, and C. Sohler. Clustering with metric and non-metric distance measures. In *SODA*, 2008.

A. Banerjee, S. Merugu, I. Dhillon, and J. Ghosh. Clustering with Bregman divergences. *Journal of Machine Learning Research*, 6:1705–1749, 2005.

S. Ben-David, U. Von Luxburg, and D. Pal. A sober look at clustering stability. In *COLT*, 2006.

G. Biau, L. Devroye, and G. Lugosi. Consistency of random forests and other averaging classifiers. *Journal of Machine Learning Research*, 9:2015–2033, 2008.

K. Chaudhuri and A. McGregor. Finding metric structure in information theoretic clustering. In *COLT*, 2008.

J. Cuesta-Albertos, A. Gordaliza, and C. Matran. Trimmed k-means: an attempt to robustify quantizers. *The Annals of Statistics*, 25(2):553–576, 1997.

I. Dhillon, S. Mallela, and R. Kumar. Divisive information-theoretic feature clustering algorithm for text classification. *Journal of Machine Learning Research*, 3:1265–1287, 2003.

T. Glasmachers. Universal consistency of multi-class support vector classification. In *NIPS*, 2010.

A. Hinneburg and D. Keim. A general approach to clustering in large databases with noise. *Knowledge and Information Systems*, 4:387–415, 2003.

S. Kpotufe, E. Sgouritsa, D. Janzing, and B. Scholkopf. Consistency of causal inference under the additive noise model. In *ICML*, 2014.

P. Long and R. Servedio. Consistency versus realizable H-consistency for multiclass classification. In *ICML*, pages 801–809, 2013.

D. McAllester and J. Keshet. Generalization bounds and consistency for latent structural probit and rump loss. In *NIPS*, 2011.

A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: analysis and an algorithm. In *NIPS*, 2001.

R. Nock, P. Luosto, and J. Kivinen. Mixed Bregman clustering with approximation guarantees. *Machine Learning and Knowledge Discovery in Databases*, pages 154–169, 2008.

D. Pollard. Strong consistency of k-means clustering. *The Annals of Statistics*, 10(1): 135–140, 1981.

A. Rakhlin and A. Caponnetto. Stability of k-means clustering. In *NIPS*, 2006.

O. Shamir and N. Tishby. Model selection and stability of k-means clustering. In *COLT*, 2008.

M. Telgarsky and S. Dasgupta. Moment-based uniform deviation bounds for k-means and friends. In *NIPS*, 2013.

V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.

U. von Luxburg, M. Belkin, and O. Bousquet. Consistency of spectral clustering. *The Annals of Statistics*, 36(2):555–586, 2008.

Journal of Machine Learning Research 16 (2015) 787-860 Submitted 5/13; Revised 9/14; Published 4/15

Risk Bounds for the Majority Vote: From a PAC-Bayesian Analysis to a Learning Algorithm

Pascal Germain Pascal.Germain@ift.ulaval.ca
Alexandre Lacasse Alexandre.Lacasse@ift.ulaval.ca
François Laviolette Francois.Laviolette@ift.ulaval.ca
Mario Marchand Mario.Marchand@ift.ulaval.ca
Jean-Francis Roy Jean-Francis.Roy@ift.ulaval.ca
Département d'informatique et de génie logiciel
Université Laval
Québec, Canada, G1V 0A6
⋆ All authors contributed equally to this work.

Editor: Koby Crammer

Abstract

We propose an extensive analysis of the behavior of majority votes in binary classification. In particular, we introduce a risk bound for majority votes, called the $\mathcal{C}$-bound, that takes into account the average quality of the voters and their average disagreement. We also propose an extensive PAC-Bayesian analysis that shows how the $\mathcal{C}$-bound can be estimated from various observations contained in the training data. The analysis intends to be self-contained and can be used as introductory material to PAC-Bayesian statistical learning theory. It starts from a general PAC-Bayesian perspective and ends with uncommon PAC-Bayesian bounds. Some of these bounds contain no Kullback-Leibler divergence and others allow kernel functions to be used as voters (via the sample compression setting). Finally, out of the analysis, we propose the MinCq learning algorithm that basically minimizes the $\mathcal{C}$-bound. MinCq reduces to a simple quadratic program. Aside from being theoretically grounded, MinCq achieves state-of-the-art performance, as shown in our extensive empirical comparison with both AdaBoost and the Support Vector Machine.

Keywords: majority vote, ensemble methods, learning theory, PAC-Bayesian theory, sample compression

1. Previous Work and Implementation

This paper can be considered as an extended version of Lacasse et al. (2006) and Laviolette et al. (2011), and also contains ideas from Laviolette and Marchand (2005, 2007) and Germain et al. (2009, 2011). We unify this previous work, revise the mathematical approach, add new results and extend empirical experiments.

The source code to compute the various PAC-Bayesian bounds presented in this paper and the implementation of the MinCq learning algorithm is available at:

`http://graal.ift.ulaval.ca/majorityvote/`

2. Introduction

In binary classification, many state-of-the-art algorithms output prediction functions that can be seen as a majority vote of "simple" classifiers. Firstly, ensemble methods such as Bagging (Breiman, 1996), Boosting (Schapire and Singer, 1999) and Random Forests (Breiman, 2001) are well-known examples of learning algorithms that output majority votes. Secondly, majority votes are also central in the Bayesian approach (see Gelman et al., 2004, for an introductory text); in this setting, the majority vote is generally called the *Bayes Classifier*. Thirdly, it is interesting to point out that classifiers produced by kernel methods, such as the Support Vector Machine (SVM) (Cortes and Vapnik, 1995), can also be viewed as majority votes. Indeed, to classify an example x, the SVM classifier computes

$$\mathrm{sgn}\left(\sum_{i=1}^{|S|}\alpha_i\, y_i\, k(x_i, x)\right), \tag{1}$$

where $k(\cdot,\cdot)$ is a kernel function, and the input-output pairs (x_i, y_i) represent the examples from the training set S. Thus, one can interpret each $y_i\, k(x_i, \cdot)$ as a voter that chooses (with confidence level $|k(x_i, x)|$) between two alternatives ("positive" or "negative"), and α_i as the respective weight of this voter in the majority vote. Then, if the total confidence-multiplied weight of each voter that votes positive is greater than the total confidence-multiplied weight of each voter that votes negative, the classifier outputs a $+1$ label (and a -1 label in the opposite case). Similarly, each *neuron* of the last layer of an artificial neural network can be interpreted as a majority vote, since it outputs a real value given by $K(\sum_i w_i g_i(x))$ for some *activation function* K.[1]

In practice, it is well known that the classifier output by each of these learning algorithms performs much better than any of its voters individually. Indeed, voting can dramatically improve performance when the "community" of classifiers tends to compensate for individual errors. In particular, this phenomenon explains the success of Boosting algorithms (*e.g.*, Schapire et al., 1998). The first aim of this paper is to explore how bounds on the generalized risk of the majority vote are not only able to theoretically justify learning algorithms but also to detect when the voted combination provably outperforms the average of its voters. We expect that this study of the behavior of a majority vote should improve the understanding of existing learning algorithms and even lead to new ones. We indeed present a learning algorithm based on these ideas at the end of the paper.

The PAC-Bayesian theory is a well-suited approach to analyze majority votes. Initiated by McAllester (1999), this theory aims to provide Probably Approximately Correct guarantees (PAC guarantees) to "Bayesian-like" learning algorithms. Within this approach, one considers a *prior*[2] *distribution* P over a space of classifiers that characterizes its prior belief about good classifiers (before the observation of the data) and a *posterior distribution* Q (over the same space of classifiers) that takes into account the additional information provided by the training data. The classical PAC-Bayesian approach indirectly bounds the risk

1. In this case, each voter g_i has incoming weights which are also learned (often by back propagation of errors) together with the weights w_i. The analysis presented in this paper considers fixed voters. Thus, the PAC-Bayesian theory for artificial neural networks remains to be done. Note however that the recent work by McAllester (2013) provides a first step in that direction.
2. Priors have been used for many years in statistics. The priors in this paper have only indirect links with the *Bayesian priors*. We nevertheless use this language, since it comes from previous work.

of a Q-weighted majority vote by bounding the risk of an associate (stochastic) classifier, called the *Gibbs classifier*. A remarkable result, known as the "PAC-Bayesian Theorem", provides a risk bound for the "true" risk of the Gibbs classifier, by considering the empirical risk of this Gibbs classifier on the training data and the Kullback-Leibler divergence between a posterior distribution Q and a prior distribution P. It is well known (Langford and Shawe-Taylor, 2002; McAllester, 2003b; Germain et al., 2009) that the risk of the (deterministic) majority vote classifier is upper-bounded by twice the risk of the associated (stochastic) Gibbs classifier. Unfortunately, and especially if the involved voters are weak, this indirect bound on the majority vote classifier is far from being tight, even if the PAC-Bayesian bound itself generally gives a tight bound on the risk of the Gibbs classifier. In practice, as stated before, the "community" of classifiers can act in such a way as to compensate for individual errors. When such compensation occurs, the risk of the majority vote is then much lower than the Gibbs risk itself and, a fortiori, much lower than twice the Gibbs risk. By limiting the analysis to Gibbs risk only, the commonly used PAC-Bayesian framework is unable to evaluate whether or not this compensation occurs. Consequently, this framework cannot help in producing highly accurate voted combinations of classifiers when these classifiers are individually weak.

In this paper, we tackle this problem by studying the margin of the majority vote as a random variable. The first and second moments of this random variable are respectively linked with the risk of the Gibbs classifier and the expected disagreement between the voters of the majority vote. As we will show, the well-known factor of two used to bound the risk of the majority vote is recovered by applying Markov's inequality to the first moment of the margin. Based on this observation, we show that a tighter bound, that we call the $\mathcal{C}$-bound, is obtained by considering the first two moments of the margin, together with Chebyshev's inequality.

Section 4 presents, in a more detailed way, the work on the $\mathcal{C}$-bound originally presented in Lacasse et al. (2006). We then present both theoretical and empirical studies that show that the $\mathcal{C}$-bound is an accurate indicator of the risk of the majority vote. We also show that the $\mathcal{C}$-bound can be smaller than the risk of the Gibbs classifier and can even be arbitrarily close to zero even if the risk of the Gibbs classifier is close to $1/2$. This indicates that the $\mathcal{C}$-bound can effectively capture the compensation of the individual errors made by the voters.

We then develop PAC-Bayesian guarantees on the $\mathcal{C}$-bound in order to obtain an upper bound on the risk of the majority vote based on empirical observations. Section 5 presents a general approach of the PAC-Bayesian theory by which we recover the most commonly used forms of the bounds of McAllester (1999, 2003a) and Langford and Seeger (2001); Seeger (2002); Langford (2005). Thereafter, we extend the theory to obtain upper bounds on the $\mathcal{C}$-bound in two different ways. The first method is to separately bound the risk of the Gibbs classifier and the expected disagreement—which are the two fundamental ingredients that are present in the $\mathcal{C}$-bound. Since the expected disagreement does not rely on labels, this strategy is well-suited for the semi-supervised learning framework. The second method directly bounds the $\mathcal{C}$-bound and empirically improves the achievable bounds in the supervised learning framework.

Sections 6 and 7 bring together relatively new PAC-Bayesian ideas that allow us, for one part, to derive a PAC-Bayesian bound that does not rely on the Kullback-Leibler divergence between the prior and posterior distributions (as in Catoni, 2007; Germain et al., 2011; Laviolette et al., 2011) and, for the other part, to extend the bound to the case where the voters are defined using elements of the training data, *e.g.*, voters defined by kernel functions $y_i k(x_i, \cdot)$. This second approach is based on the sample compression theory (Floyd and Warmuth, 1995; Laviolette and Marchand, 2007; Germain et al., 2011). In PAC-Bayesian theory, the sample compression approach is *a priori* problematic, since a PAC-Bayesian bound makes use of a prior distribution on the set of all voters that has to be defined before observing the data. If the voters themselves are defined using a part of the data, there is an apparent contradiction that has to be overcome.

Based on the foregoing, a learning algorithm, that we call MinCq, is presented in Section 8. The algorithm basically minimizes the $\mathcal{C}$-bound, but in a particular way that is, inter alia, justified by the PAC-Bayesian analysis of Sections 6 and 7. This algorithm was originally presented in Laviolette et al. (2011). Given a set of voters (either classifiers or kernel functions), MinCq builds a majority vote classifier by finding the posterior distribution Q on the set of voters that minimizes the $\mathcal{C}$-bound. Hence, MinCq takes into account not only the overall quality of the voters, but also their respective disagreements. In this way, MinCq builds a "community" of voters that can compensate for their individual errors. Even though the $\mathcal{C}$-bound consists of a relatively complex quotient, the MinCq learning algorithm reduces to a simple quadratic program. Moreover, extensive empirical experiments confirm that MinCq is very competitive when compared with AdaBoost (Schapire and Singer, 1999) and the Support Vector Machine (Cortes and Vapnik, 1995).

In Section 9, we conclude by pointing out recent work that uses the PAC-Bayesian theory to tackle more sophisticated machine learning problems.

3. Basic Definitions

We consider classification problems where the input space $\mathcal{X}$ is an arbitrary set and the output space is a discrete set denoted $\mathcal{Y}$. An *example* (x, y) is an input-output pair where $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. A *voter* is a function $\mathcal{X} \to \overline{\mathcal{Y}}$ for some output space $\overline{\mathcal{Y}}$ related to $\mathcal{Y}$. Unless otherwise specified, we consider the binary classification problem where $\mathcal{Y} = \{-1, 1\}$ and then we either consider $\overline{\mathcal{Y}}$ as $\mathcal{Y}$ itself, or its convex hull $[-1, +1]$. In this paper, we also use the following convention: f denotes a real-valued voter (*i.e.*, $\overline{\mathcal{Y}} = [-1, 1]$), and h denotes a binary-valued voter (*i.e.*, $\overline{\mathcal{Y}} = \{-1, 1\}$). Note that this notion of voters is quite general, since any uniformly bounded real-valued set of functions can be viewed as a set of voters when properly normalized.

We consider learning algorithms that construct majority votes based on a (finite) set $\mathcal{H}$ of voters. Given any $x \in \mathcal{X}$, the output $B_Q(x)$ of a Q-weighted majority vote classifier B_Q (sometimes called the *Bayes classifier*) is given by

$$B_Q(x) \stackrel{\text{def}}{=} \operatorname{sgn}\left[\mathop{\mathbf{E}}_{f \sim Q} f(x)\right], \tag{2}$$

where $\operatorname{sgn}(a) = 1$ if $a > 0$, $\operatorname{sgn}(a) = -1$ if $a < 0$, and $\operatorname{sgn}(0) = 0$.

Thus, in case of a tie in the majority vote – *i.e.*, $\mathbf{E}_{f\sim Q} f(x) = 0$ –, we consider that the majority vote classifier abstains – *i.e.*, $B_Q(x) = 0$. There are other possible ways to handle this particular case. In this paper, we choose to define $\mathrm{sgn}(0) = 0$ because it simplifies the forthcoming analysis.

We adopt the PAC setting where each example (x, y) is drawn i.i.d. according to a fixed, but unknown, probability distribution D on $\mathcal{X} \times \mathcal{Y}$. The *training set* of m examples is denoted by $S = \langle (x_1, y_1), \ldots, (x_m, y_m) \rangle \sim D^m$. Throughout the paper, D' generically represents either the true (and unknown) distribution D, or its empirical counterpart U_S (*i.e.*, the uniform distribution over the training set S). Moreover, for notational simplicity, we often replace U_S by S.

In order to quantify the accuracy of a voter, we use a *loss function* $\mathcal{L} : \overline{\mathcal{Y}} \times \mathcal{Y} \to [0, 1]$. The PAC-Bayesian theory traditionally considers majority votes of binary voters of the form $h : \mathcal{X} \to \{-1, 1\}$, and the *zero-one loss* $\mathcal{L}_{01}\big(h(x), y\big) \stackrel{\text{def}}{=} I\big(h(x) \neq y\big)$, where $I(a) = 1$ if predicate a is true and 0 otherwise.

The extension of the zero-one loss to real-valued voters (of the form $f : \mathcal{X} \to [-1, 1]$) is given by the following definition.

Definition 1 In the (more general) case where voters are functions $f : \mathcal{X} \to [-1, 1]$, the zero-one loss $\mathcal{L}_{01}$ is defined by

$$\mathcal{L}_{01}\big(f(x), y\big) \stackrel{\text{def}}{=} I\big(y \cdot f(x) \leq 0\big).$$

Hence, a voter abstention – *i.e.*, when $f(x)$ outputs exactly 0 – results in a loss of 1. Clearly, other choices are possible for this particular case.[3]

In this paper, we also consider the *linear loss* $\mathcal{L}_\ell$ defined as follows.

Definition 2 Given a voter $f : \mathcal{X} \to [-1, 1]$, the linear loss $\mathcal{L}_\ell$ is defined by

$$\mathcal{L}_\ell\big(f(x), y\big) \stackrel{\text{def}}{=} \frac{1}{2}\Big(1 - y \cdot f(x)\Big).$$

Note that the linear loss is equal to the zero-one loss when the output space is binary. That is, for any $(h(x), y) \in \{-1, 1\}^2$, we always have

$$\mathcal{L}_\ell\big(h(x), y\big) \;=\; \mathcal{L}_{01}\big(h(x), y\big)\,, \tag{3}$$

because $\mathcal{L}_\ell\big(h(x), y\big) = 1$ if $h(x) \neq y$, and $\mathcal{L}_\ell\big(h(x), y\big) = 0$ if $h(x) = y$. Hence, we generalize all definitions implying classifiers to voters using the equality of Equation (3) as an inspiration. Figure 1 illustrates the difference between the zero-one loss and the linear loss for real-valued voters. Remember that in the case $y\, f(x) = 0$, the loss is 1 (see Definition 1).

Definition 3 Given a loss function $\mathcal{L}$ and a voter f, the *expected loss* $\mathbb{E}^{\mathcal{L}}_{D'}(f)$ of f relative to distribution D' is defined as

$$\mathbb{E}^{\mathcal{L}}_{D'}(f) \stackrel{\text{def}}{=} \mathop{\mathbf{E}}_{(x,y)\sim D'} \mathcal{L}\big(f(x), y\big)\,.$$

3. As an example, when $f(x)$ outputs 0, the loss may be 1/2. However, we choose for this unlikely event the worst loss value – *i.e.*, $\mathcal{L}_{01}(0, y) = 1$ – because it simplifies the majority vote analysis.

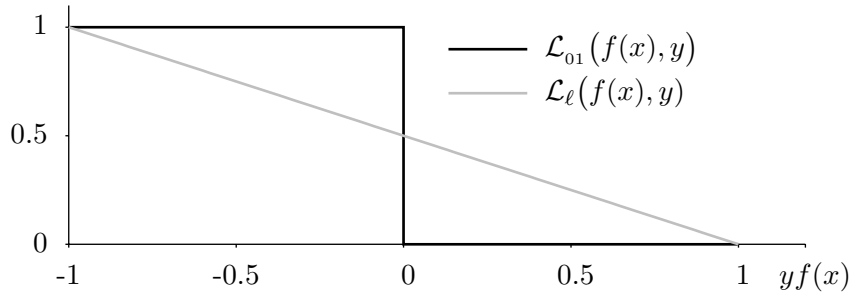


Figure 1: The zero-one loss $\mathcal{L}_{01}$ and the linear loss $\mathcal{L}_{\ell}$ as a function of $yf(x)$.

In particular, the *empirical expected loss* on a training set S is given by

$$\mathbb{E}_S^{\mathcal{L}}(f) \;=\; \frac{1}{m}\sum_{i=1}^{m}\mathcal{L}\big(f(x_i),y_i\big)\,.$$

We therefore define the risk of the majority vote $R_{D'}(B_Q)$ as follows.

Definition 4 For any probability distribution Q on a set of voters, the *Bayes risk* $R_{D'}(B_Q)$, also called *risk of the majority vote*, is defined as the expected zero-one loss of the majority vote classifier B_Q relative to D'. Hence,

$$R_{D'}(B_Q) \;\stackrel{\text{def}}{=}\; \mathbb{E}_{D'}^{\mathcal{L}_{01}}(B_Q) \;=\; \mathop{\mathbf{E}}_{(x,y)\sim D'} I\Big(B_Q(x)\neq y\Big) \;=\; \mathop{\mathbf{E}}_{(x,y)\sim D'} I\Big(\mathop{\mathbf{E}}_{f\sim Q} y\cdot f(x)\leq 0\Big)\,.$$

Remember from the definition of B_Q (Equation 2) that the majority vote classifier abstains in the case of a tie on an example (x,y). Therefore, the above Definition 4 implies that the Bayes risk is 1 in this case, as $R_{\langle(x,y)\rangle}(B_Q)=\mathcal{L}_{01}(0,y)=1$. In practice, a tie in the vote is a rare event, especially if there are many voters.

The output of the deterministic majority vote classifier B_Q is closely related to the output of a stochastic classifier called the *Gibbs classifier*. To classify an input example x, the Gibbs classifier G_Q randomly chooses a voter f according to Q and returns $f(x)$. Note the stochasticity of the Gibbs classifier: it can output different values when given the same input x twice. We will see later how the link between B_Q and G_Q is used in the PAC-Bayesian theory.

In the case of binary voters, the Gibbs risk corresponds to the probability that G_Q misclassifies an example of distribution D'. Hence,

$$R_{D'}(G_Q) \;=\; \Pr_{\substack{(x,y)\sim D'\\ h\sim Q}}\big(h(x)\neq y\big) \;=\; \mathop{\mathbf{E}}_{h\sim Q}\mathbb{E}_{D'}^{\mathcal{L}_{01}}(h) \;=\; \mathop{\mathbf{E}}_{(x,y)\sim D'}\mathop{\mathbf{E}}_{h\sim Q} I\big(h(x)\neq y\big)\,.$$

In order to handle real-valued voters, we generalize the Gibbs risk as follows.

Definition 5 For any probability distribution Q on a set of voters, the *Gibbs risk* $R_{D'}(G_Q)$ is defined as the expected linear loss of the Gibbs classifier G_Q relative to D'. Hence,

$$R_{D'}(G_Q) \;\stackrel{\text{def}}{=}\; \mathop{\mathbf{E}}_{f\sim Q}\mathbb{E}_{D'}^{\mathcal{L}_{\ell}}(f) \;=\; \frac{1}{2}\left(1-\mathop{\mathbf{E}}_{(x,y)\sim D'}\mathop{\mathbf{E}}_{f\sim Q} y\cdot f(x)\right)\,.$$

Remark 6 It is well known in the PAC-Bayesian literature (*e.g.*, Langford and Shawe-Taylor, 2002; McAllester, 2003b; Germain et al., 2009) that the Bayes risk $R_{D'}(B_Q)$ is bounded by twice the Gibbs risk $R_{D'}(G_Q)$. This statement extends to our more general definition of the Gibbs risk (Definition 5).

Proof Let $(x, y) \in \mathcal{X} \times \{-1, 1\}$ be any example. We claim that

$$R_{\langle(x,y)\rangle}(B_Q) \;\leq\; 2\, R_{\langle(x,y)\rangle}(G_Q)\,. \tag{4}$$

Notice that $R_{\langle(x,y)\rangle}(B_Q)$ is either 0 or 1 depending of the fact that B_Q errs or not on (x, y). In the case where $R_{\langle(x,y)\rangle}(B_Q) = 0$, Equation (4) is trivially true. If $R_{\langle(x,y)\rangle}(B_Q) = 1$, we know by the last equality of Definition 4 that $\mathop{\mathbf{E}}_{f\sim Q} y \cdot f(x) \leq 0$. Therefore, Definition 5 gives

$$2 \cdot R_{\langle(x,y)\rangle}(G_Q) \;=\; 2 \cdot \frac{1}{2}\left(1 - \mathop{\mathbf{E}}_{f\sim Q} y \cdot f(x)\right) \;\geq\; 1 = R_{\langle(x,y)\rangle}(B_Q)\,,$$

which proves the claim.

Now, by taking the expectation according to $(x, y) \sim D'$ on each side of Equation (4), we obtain

$$R_{D'}(B_Q) \;=\; \mathop{\mathbf{E}}_{(x,y)\sim D'} R_{\langle(x,y)\rangle}(B_Q) \;\leq\; \mathop{\mathbf{E}}_{(x,y)\sim D'} 2\, R_{\langle(x,y)\rangle}(G_Q) \;= 2\, R_{D'}(G_Q)\,,$$

as wanted. ■

Thus, PAC-Bayesian bounds on the risk of the majority vote are usually bounds on the Gibbs risk, multiplied by a factor of two. Even if this type of bound can be tight in some situations, the factor two can also be misleading. Langford and Shawe-Taylor (2002) have shown that under some circumstances, the factor of two can be reduced to $(1 + \epsilon)$. Nevertheless, distributions Q on voters giving $R_{D'}(G_Q) \gg R_{D'}(B_Q)$ are common. The extreme case happens when the expected linear loss on each example is just below one half – *i.e.*, for all (x, y), $\mathbf{E}_{f\sim Q}\, y\, f(x) = \frac{1}{2} - \epsilon$ –, leading to a perfect majority vote classifier but an almost inaccurate Gibbs classifier. Indeed, we have $R_{D'}(G_Q) = \frac{1}{2} - \epsilon$ and $R_{D'}(B_Q) = 0$. Therefore, in this circumstance, the bound $R_{D'}(B_Q) \leq 1 - 2\epsilon$, given by Remark 6, fails to represent the perfect accuracy of the majority vote. This problem is due to the fact that the Gibbs risk only considers the loss of the average output of the population of voters. Hence, the bound of Remark 6 states that the majority vote is weak whenever every individual voter is weak. The bound cannot capture the fact that it might happen that the "community" of voters compensates for individual errors. To overcome this lacuna, we need a bound that compares the output of voters between them, not only the average quality of each voter taken individually.

We can compare the output of binary voters by considering the probability of disagreement between them:

$$\begin{aligned}
\Pr_{\substack{x\sim D'_{\mathcal{X}} \\ h_1, h_2 \sim Q}}\Big(h_1(x) \neq h_2(x)\Big) \;&=\; \mathop{\mathbf{E}}_{x\sim D'_{\mathcal{X}}} \mathop{\mathbf{E}}_{h_1\sim Q} \mathop{\mathbf{E}}_{h_2\sim Q} I\Big(h_1(x) \neq h_2(x)\Big) \\
&=\; \mathop{\mathbf{E}}_{x\sim D'_{\mathcal{X}}} \mathop{\mathbf{E}}_{h_1\sim Q} \mathop{\mathbf{E}}_{h_2\sim Q} I\Big(h_1(x) \cdot h_2(x) \neq 1\Big) \\
&=\; \mathop{\mathbf{E}}_{x\sim D'_{\mathcal{X}}} \mathop{\mathbf{E}}_{h_1\sim Q} \mathop{\mathbf{E}}_{h_2\sim Q} \mathcal{L}_{01}\big(\, h_1(x) \cdot h_2(x)\,,\, 1\,\big)\,,
\end{aligned}$$

where $D'_{\mathcal{X}}$ denotes the marginal on $\mathcal{X}$ of distribution D'. Definition 7 extends this notion of disagreement to real-valued voters.

Definition 7 For any probability distribution Q on a set of voters, the *expected disagreement* $d_Q^{D'}$ relative to D' is defined as

$$\begin{aligned} d_Q^{D'} &\stackrel{\text{def}}{=} \mathop{\mathbf{E}}_{x\sim D'_{\mathcal{X}}} \mathop{\mathbf{E}}_{f_1\sim Q} \mathop{\mathbf{E}}_{f_2\sim Q} \mathcal{L}_\ell\left(f_1(x)\cdot f_2(x)\,,\,1\right) \\ &= \frac{1}{2}\left(1 - \mathop{\mathbf{E}}_{x\sim D'_{\mathcal{X}}} \mathop{\mathbf{E}}_{f_1\sim Q} \mathop{\mathbf{E}}_{f_2\sim Q} 1\cdot f_1(x)\cdot f_2(x)\right) \\ &= \frac{1}{2}\left(1 - \mathop{\mathbf{E}}_{x\sim D'_{\mathcal{X}}} \left[\mathop{\mathbf{E}}_{f\sim Q} f(x)\right]^2\right). \end{aligned}$$

Notice that the value of $d_Q^{D'}$ does not depend on the labels y of the examples $(x,y)\sim D'$. Therefore, we can estimate the expected disagreement with unlabeled data.

4. Bounds on the Risk of the Majority Vote

The aim of this section is to introduce the $\mathcal{C}$-bound, which upper-bounds the risk of the majority vote (Definition 4) based on the Gibbs risk (Definition 5) and the expected disagreement (Definition 7). We start by studying the margin of a majority vote as a random variable (Section 4.1). From the first moment of the margin, we easily recover the well-known bound of twice the Gibbs risk presented by Remark 6 (Section 4.2). We therefore suggest extending this analysis to the second moment of the margin to obtain the $\mathcal{C}$-bound (Section 4.3). Finally, we present some statistical properties of the $\mathcal{C}$-bound (Section 4.4) and an empirical study of its predictive power (Section 4.5).

4.1 The Margin of the Majority Vote and its Moments

The bounds on the risk of a majority vote classifier proposed in this section result from the study of the weighted margin of the majority vote as a random variable.

Definition 8 Let $M_Q^{D'}$ be the random variable that, given any example (x,y) drawn according to D', outputs the *margin* of the majority vote B_Q on that example, which is

$$M_Q(x,y) \stackrel{\text{def}}{=} \mathop{\mathbf{E}}_{f\sim Q} y\cdot f(x)\,.$$

From Definitions 4 and 8, we have the following nice property:[4]

$$R_{D'}(B_Q) \;=\; \Pr_{(x,y)\sim D'}\Big(M_Q(x,y)\le 0\Big)\,. \tag{5}$$

4. Note that for another choice of the zero-one loss definition (Definition 1), the tie in the majority vote – *i.e.*, when $M_Q(x,y)=0$ – would have been more complicated to handle, and the statement should have been relaxed to

$$\Pr_{(x,y)\sim D'}\Big(M_Q(x,y)<0\Big) \;\le\; R_{D'}(B_Q) \;\le\; \Pr_{(x,y)\sim D'}\Big(M_Q(x,y)\le 0\Big)\,.$$

The margin is not only related to the risk of the majority vote, but also to Gibbs risk. For that purpose, let us consider the first moment $\mu_1(M_Q^{D'})$ of the random variable $M_Q^{D'}$ which is defined as

$$\mu_1(M_Q^{D'}) \stackrel{\text{def}}{=} \mathop{\mathbf{E}}_{(x,y)\sim D'} M_Q(x,y)\,. \tag{6}$$

We can now rewrite the Gibbs risk (Definition 5) as a function of $\mu_1(M_Q^{D'})$, since

$$\begin{aligned} R_{D'}(G_Q) &= \mathop{\mathbf{E}}_{f\sim Q} \mathbb{E}_{D'}^{\mathcal{L}_\ell}(f) \;=\; \frac{1}{2}\left(1 - \mathop{\mathbf{E}}_{(x,y)\sim D'} \mathop{\mathbf{E}}_{f\sim Q} y\cdot f(x)\right) \\ &= \frac{1}{2}\Big(1 - \mathop{\mathbf{E}}_{(x,y)\sim D'} M_Q(x,y)\Big) \\ &= \frac{1}{2}\Big(1 - \mu_1(M_Q^{D'})\Big)\,. \end{aligned} \tag{7}$$

Similarly, we can rewrite the expected disagreement as a function of the second moment of the margin. We use $\mu_2(M_Q^{D'})$ to denote the second moment. Since $y \in \{-1,1\}$ and, therefore, $y^2 = 1$, the second moment of the margin does not rely on labels. Indeed, we have

$$\begin{aligned} \mu_2(M_Q^{D'}) &\stackrel{\text{def}}{=} \mathop{\mathbf{E}}_{(x,y)\sim D'} \Big[M_Q(x,y)\Big]^2 \\ &= \mathop{\mathbf{E}}_{(x,y)\sim D'} y^2 \cdot \Big[\mathop{\mathbf{E}}_{f\sim Q} f(x)\Big]^2 \\ &= \mathop{\mathbf{E}}_{x\sim D'_{\mathcal{X}}} \Big[\mathop{\mathbf{E}}_{f\sim Q} f(x)\Big]^2 . \end{aligned} \tag{8}$$

Hence, from the last equality and Definition 7, the expected disagreement can be expressed as

$$\begin{aligned} d_Q^{D'} &= \frac{1}{2}\left(1 - \mathop{\mathbf{E}}_{x\sim D'_{\mathcal{X}}} \left[\mathop{\mathbf{E}}_{f\sim Q} f(x)\right]^2\right) \\ &= \frac{1}{2}\Big(1 - \mu_2(M_Q^{D'})\Big)\,. \end{aligned} \tag{9}$$

Equation (9) shows that $0 \le d_Q^{D'} \le 1/2$, since $0 \le \mu_2(M_Q^{D'}) \le 1$. Furthermore, we can upper-bound the disagreement more tightly than simply saying it is at most $1/2$ by making use of the value of the Gibbs risk. To do so, let us write the variance of the margin as

$$\begin{aligned} \mathrm{Var}(M_Q^{D'}) &\stackrel{\text{def}}{=} \mathop{\mathbf{Var}}_{(x,y)\sim D'} \big(M_Q(x,y)\big) \\ &= \mu_2(M_Q^{D'}) - \big(\mu_1(M_Q^{D'})\big)^2 . \end{aligned} \tag{10}$$

Therefore, as the variance cannot be negative, it follows that

$$\mu_2(M_Q^{D'}) \;\ge\; \big(\mu_1(M_Q^{D'})\big)^2 ,$$

which implies that

$$1 - 2 \cdot d_Q^{D'} \;\geq\; \left(1 - 2 \cdot R_{D'}(G_Q)\right)^2 . \tag{11}$$

Easy calculation then gives the desired bound of $d_Q^{D'}$ (that is based on the Gibbs risk):

$$d_Q^{D'} \;\leq\; 2 \cdot R_{D'}(G_Q) \cdot \left(1 - R_{D'}(G_Q)\right). \tag{12}$$

We therefore have the following proposition.

Proposition 9 *For any distribution Q on a set of voters and any distribution D' on $\mathcal{X} \times \{-1, 1\}$, we have*

$$d_Q^{D'} \;\leq\; 2 \cdot R_{D'}(G_Q) \cdot \left(1 - R_{D'}(G_Q)\right) \;\leq\; \frac{1}{2}.$$

Moreover, if $d_Q^{D'} = \frac{1}{2}$ then $R_{D'}(G_Q) = \frac{1}{2}$.

Proof Equation (12) gives the first inequality. The rest of the proposition directly follows from the fact that $f(x) = 2x(1-x)$ is a parabola whose (unique) maximum is at the point $(\frac{1}{2}, \frac{1}{2})$. ■

4.2 Rediscovering the bound $R_{D'}(B_Q) \leq 2 \cdot R_{D'}(G_Q)$

The well-known factor of two with which one can transform a bound on the Gibbs risk $R_{D'}(G_Q)$ into a bound on the risk $R_{D'}(B_Q)$ of the majority vote is usually justified by an argument similar to the one given in Remark 6. However, as shown by the proof of Proposition 10, the result can also be obtained by considering that the risk of the majority vote is the probability that the margin $M_Q^{D'}$ is lesser than or equal to zero (Equation 5) and by simply applying Markov's inequality (Lemma 46, provided in Appendix A).

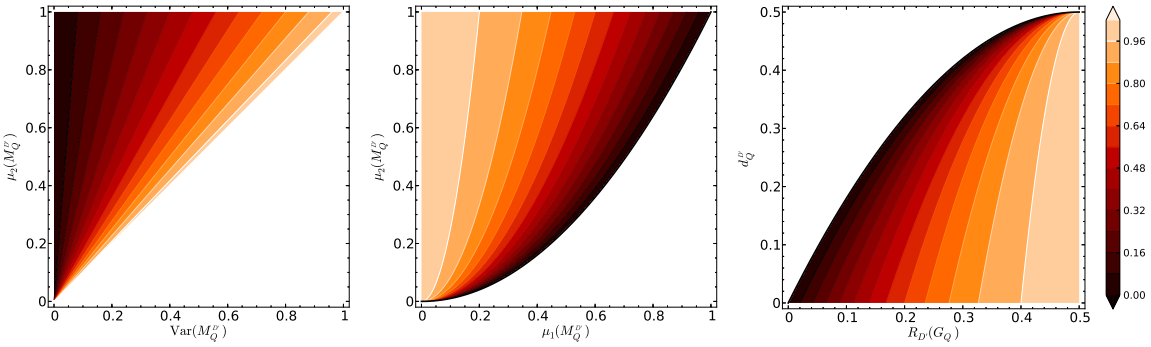
Proposition 10 *For any distribution Q on a set of voters and any distribution D' on $\mathcal{X} \times \{-1, 1\}$, we have*

$$R_{D'}(B_Q) \;\leq\; 2 \cdot R_{D'}(G_Q).$$

Proof Starting from Equation (5) and using Markov's inequality (Lemma 46), we have

$$\begin{aligned}
R_{D'}(B_Q) &= \Pr_{(x,y)\sim D'}\left(M_Q(x,y) \leq 0\right) \\
&= \Pr_{(x,y)\sim D'}\left(1 - M_Q(x,y) \geq 1\right) \\
&\leq \mathop{\mathbf{E}}_{(x,y)\sim D'}\left(1 - M_Q(x,y)\right) && \text{(Markov's inequality)} \\
&= 1 - \mathop{\mathbf{E}}_{(x,y)\sim D'} M_Q(x,y) \\
&= 1 - \mu_1(M_Q^{D'}) \\
&= 2 \cdot R_{D'}(G_Q).
\end{aligned}$$

The last equality is directly obtained from Equation (7). ■



(a) First form. (b) Second form. (c) Third form.

Figure 2: Contour plots of the $\mathcal{C}$-bound.

This proof highlights that we can upper-bound $R_{D'}(B_Q)$ by considering solely the first moment of the margin $\mu_1(M_Q^{D'})$. Once we realize this fact, it becomes natural to extend this result to higher moments. We do so in the following subsection where we make use of Chebyshev's inequality (instead of Markov's inequality), which uses not only the first, but also the second moment of the margin. This gives rise to the $\mathcal{C}$-bound of Theorem 11.

4.3 The $\mathcal{C}$-bound: a Bound on $R_{D'}(B_Q)$ That Can Be Much Smaller Than $R_{D'}(G_Q)$

Here is the bound on which most of the results of this paper are based. We refer to it as the $\mathcal{C}$-bound. It was first introduced (but in a different form) in Lacasse et al. (2006).[5] We give here three different (but equivalent) forms of the $\mathcal{C}$-bound. Each one highlights a different property or behavior of the bound. Figure 2 illustrates these behaviors.

It is interesting to note that the proof of Theorem 11 below has the same starting point as the proof of Proposition 10, but uses Chebyshev's inequality instead of Markov's inequality (respectively Lemmas 48 and 46, both provided in Appendix A). Therefore, Theorem 11 is based on the variance of the margin in addition of its mean.

Theorem 11 (The $\mathcal{C}$-bound) *For any distribution Q on a set of voters and any distribution D' on $\mathcal{X}\times\{-1,1\}$, if $\mu_1(M_Q^{D'})>0$ (i.e., $R_{D'}(G_Q)<1/2$), we have*

$$R_{D'}(B_Q) \;\leq\; \mathcal{C}_Q^{D'}\,,$$

where

$$\mathcal{C}_Q^{D'} \;\stackrel{\text{def}}{=}\; \underbrace{\frac{\mathrm{Var}(M_Q^{D'})}{\mu_2(M_Q^{D'})}}_{\textit{First form}} \;=\; \underbrace{1-\frac{\left(\mu_1(M_Q^{D'})\right)^2}{\mu_2(M_Q^{D'})}}_{\textit{Second form}} \;=\; \underbrace{1-\frac{\left(1-2\cdot R_{D'}(G_Q)\right)^2}{1-2\cdot d_Q^{D'}}}_{\textit{Third form}}\,.$$

5. We present the form used by Lacasse et al. (2006) in Remark 12 at the end of the present subsection.

Proof Starting from Equation (5) and using the one-sided Chebyshev inequality (Lemma 48), with $X = -M_Q(x,y)$, $\mu = \mathop{\mathbf{E}}_{(x,y)\sim D'}\big(-M_Q(x,y)\big)$ and $a = \mathop{\mathbf{E}}_{(x,y)\sim D'} M_Q(x,y)$, we obtain

$$
\begin{aligned}
R_{D'}(B_Q) &= \Pr_{(x,y)\sim D'}\Big(M_Q(x,y) \leq 0\Big) \\
&= \Pr_{(x,y)\sim D'}\left(-M_Q(x,y) + \mathop{\mathbf{E}}_{(x,y)\sim D'} M_Q(x,y) \geq \mathop{\mathbf{E}}_{(x,y)\sim D'} M_Q(x,y)\right) \\
&\leq \frac{\mathop{\mathbf{Var}}_{(x,y)\sim D'}\left(M_Q(x,y)\right)}{\mathop{\mathbf{Var}}_{(x,y)\sim D'}\left(M_Q(x,y)\right) + \left(\mathop{\mathbf{E}}_{(x,y)\sim D'} M_Q(x,y)\right)^2} \qquad \text{(Chebyshev's inequality)} \\
&= \frac{\mathrm{Var}(M_Q^{D'})}{\mu_2(M_Q^{D'}) - \left(\mu_1(M_Q^{D'})\right)^2 + \left(\mu_1(M_Q^{D'})\right)^2} = \frac{\mathrm{Var}(M_Q^{D'})}{\mu_2(M_Q^{D'})} \qquad (13) \\
&= \frac{\mu_2(M_Q^{D'}) - \left(\mu_1(M_Q^{D'})\right)^2}{\mu_2(M_Q^{D'})} \\
&= 1 - \frac{\left(\mu_1(M_Q^{D'})\right)^2}{\mu_2(M_Q^{D'})} \qquad (14) \\
&= 1 - \frac{\left(1 - 2\cdot R_{D'}(G_Q)\right)^2}{1 - 2\cdot d_Q^{D'}}\,. \qquad (15)
\end{aligned}
$$

Lines (13) and (14) respectively present the first and the second forms of $\mathcal{C}_Q^{D'}$, and follow from the definitions of $\mu_1(M_Q^{D'})$, $\mu_2(M_Q^{D'})$, and $\mathrm{Var}(M_Q^{D'})$ (see Equations 6, 8 and 10). The third form of $\mathcal{C}_Q^{D'}$ is obtained at Line (15) using $\mu_1(M_Q^{D'}) = 1 - 2\cdot R_{D'}(G_Q)$ and $\mu_2(M_Q^{D'}) = 1 - 2\cdot d_Q^{D'}$, which can be derived directly from Equations (7) and (9). ■

The third form of the $\mathcal{C}$-bound shows that the bound decreases when the Gibbs risk $R_{D'}(G_Q)$ decreases or when the disagreement $d_Q^{D'}$ increases. This new bound therefore suggests that a majority vote should perform a trade-off between the Gibbs risk and the disagreement in order to achieve a low Bayes risk. This is more informative than the usual bound of Proposition 10, which focuses solely on the minimization of the Gibbs risk.

The first form of the $\mathcal{C}$-bound highlights that its value is always positive (since the variance and the second moment of the margin are positive), whereas the second form of the $\mathcal{C}$-bound highlights that it cannot exceed one. Finally, the fact that $d_Q^{D'} = \frac{1}{2} \Rightarrow R_{D'}(G_Q) = \frac{1}{2}$ (Proposition 9) implies that the bound is always defined, since $R_{D'}(G_Q)$ is here assumed to be strictly less than $\frac{1}{2}$.

Remark 12 As explained before, the $\mathcal{C}$-bound was originally stated in Lacasse et al. (2006), but in a different form. It was presented as a function of $W_Q(x,y)$, the Q-weight of voters making an error on example (x,y). More precisely, the $\mathcal{C}$-bound was presented as follows:

$$
\mathcal{C}_Q^D = \frac{\mathop{\mathbf{Var}}_{(x,y)\sim D'}\big(W_Q(x,y)\big)}{\mathop{\mathbf{Var}}_{(x,y)\sim D'}\big(W_Q(x,y)\big) + \left(1/2 - R_{D'}(G_Q)\right)^2}\,.
$$

It is easy to show that this form is equivalent to the three forms stated in Theorem 11, and that $W_Q(x,y)$ and $M_Q(x,y)$ are related by

$$W_Q(x,y) \stackrel{\text{def}}{=} \mathop{\mathbf{E}}_{f\sim Q} \mathcal{L}_\ell\big(f(x),y\big) \;=\; \frac{1}{2}\left(1 - y\cdot \mathop{\mathbf{E}}_{f\sim Q} f(x)\right) \;=\; \frac{1}{2}\Big(1 - M_Q(x,y)\Big)\,.$$

However, we do not discuss further this form of the $\mathcal{C}$-bound here, since we now consider that the margin $M_Q(x,y)$ is a more natural notion than $W_Q(x,y)$.

4.4 Statistical Analysis of the $\mathcal{C}$-bound's Behavior

This section presents some properties of the $\mathcal{C}$-bound. In the first place, we discuss the conditions under which the $\mathcal{C}$-bound is optimal, in the sense that if the only information that one has about a majority vote is the first two moments of its margin distribution, it is possible that the value given by the $\mathcal{C}$-bound *is* the Bayes risk, *i.e.*, $\mathcal{C}_Q^{D'} = R_{D'}(B_Q)$.[6] In the second place, we show that the $\mathcal{C}$-bound can be arbitrarily small, especially in the presence of "non-correlated" voters, even if the Gibbs risk is large, *i.e.*, $\mathcal{C}_Q^{D'} \ll R_{D'}(G_Q)$.

4.4.1 Conditions of Optimality

For the sake of simplicity, let us focus on a random variable M that represents a margin distribution (here, we ignore underlying distributions Q on $\mathcal{H}$ and D' on $\mathcal{X}\times\{-1,1\}$) of first moment $\mu_1(M)$ and second moment $\mu_2(M)$. By Equation (5), we have

$$R(B_M) \;\stackrel{\text{def}}{=}\; \Pr\left(M \leq 0\right). \tag{16}$$

Moreover, $R(B_M)$ is upper-bounded by $\mathcal{C}_M$, the $\mathcal{C}$-bound given by the second form of Theorem 11,

$$\mathcal{C}_M \;\stackrel{\text{def}}{=}\; 1 - \frac{\big(\,\mu_1(M)\,\big)^2}{\mu_2(M)}\,. \tag{17}$$

The next proposition shows when the $\mathcal{C}$-bound can be achieved.

Proposition 13 (Optimality of the $\mathcal{C}$-bound) *Let M be any random variable that represents the margin of a majority vote. Then there exists a random variable $\widetilde{M}$ such that*

$$\mu_1(\widetilde{M}) = \mu_1(M)\,, \quad \mu_2(\widetilde{M}) = \mu_2(M)\,, \quad \textit{and} \quad \mathcal{C}_{\widetilde{M}} = \mathcal{C}_M = R(B_{\widetilde{M}}) \tag{18}$$

if and only if

$$0 < \mu_2(M) \leq \mu_1(M)\,. \tag{19}$$

Proof First, let us show that (19) implies (18). Given $0 < \mu_2(M) \leq \mu_1(M)$, we consider a distribution $\widetilde{M}$ concentrated in two points defined as

$$\widetilde{M} \;=\; \begin{cases} 0 & \text{with probability } \mathcal{C}_M = 1 - \dfrac{\big(\mu_1(M)\big)^2}{\mu_2(M)}\,, \\[2ex] \dfrac{\mu_2(M)}{\mu_1(M)} & \text{with probability } 1-\mathcal{C}_M = \dfrac{\big(\mu_1(M)\big)^2}{\mu_2(M)}\,. \end{cases}$$

6. In other words, the *optimality of the $\mathcal{C}$-bound* means here that there exists a random variable with the same first moments as the margin distribution, such that Chebyshev's inequality of Lemma 48 is reached.

This distribution has the required moments, as

$$\mu_1(\widetilde{M}) = \frac{\big(\mu_1(M)\big)^2}{\mu_2(M)}\left[\frac{\mu_2(M)}{\mu_1(M)}\right] = \mu_1(M)\,, \text{ and } \mu_2(\widetilde{M}) = \frac{\big(\mu_1(M)\big)^2}{\mu_2(M)}\left[\frac{\mu_2(M)}{\mu_1(M)}\right]^2 = \mu_2(M)\,.$$

It follows directly from Equation (17) that $\mathcal{C}_{\widetilde{M}} = \mathcal{C}_M$. Moreover, by Equation (16) and because $\frac{\mu_2(M)}{\mu_1(M)} > 0$, we obtain as desired

$$R(B_{\widetilde{M}}) \;=\; \Pr\,(\widetilde{M} \leq 0) \;=\; \mathcal{C}_M\,.$$

Now, let us show that (18) implies (19). Consider a distribution $\widetilde{M}$ such that the equalities of Line (18) are satisfied. By Proposition 10 and Equation (7), we obtain the inequality

$$\mathcal{C}_M \;=\; R(B_{\widetilde{M}}) \;\leq\; 1 - \mu_1(\widetilde{M}) \;=\; 1 - \mu_1(M)\,.$$

Hence, by the definition of $\mathcal{C}_M$, we have

$$1 - \frac{\big(\,\mu_1(M)\,\big)^2}{\mu_2(M)} \;\leq\; 1 - \mu_1(M)\,,$$

which, by straightforward calculations, implies $0 \;<\; \mu_2(M) \;\leq\; \mu_1(M)$, and we are done. ■

We discussed in Section 4.1 the multiple connections between the moments of the margin, the Gibbs risk and the expected disagreement of a majority vote. In the next proposition, we exploit these connections to derive expressions equivalent to Line (19) of Proposition 13. Thus, this shows three (equivalent) necessary conditions under which the $\mathcal{C}$-bound is optimal.

Proposition 14 *For any distribution Q on a set of voters and any distribution D' on $\mathcal{X}\times\{-1,1\}$, if $\mu_1(M_Q^{D'}) > 0$ (i.e., $R_{D'}(G_Q) < 1/2$), then the three following statements are equivalent:*

(i) $\mu_2(M_Q^{D'}) \;\leq\; \mu_1(M_Q^{D'})$;

(ii) $R_{D'}(G_Q) \;\leq\; d_Q^{D'}$;

(iii) $\mathcal{C}_Q^{D'} \;\leq\; 2\,R_{D'}(G_Q)$.

Proof The truth of $(i) \Leftrightarrow (ii)$ is a direct consequence of Equations (7) and (9). To prove $(ii) \Leftrightarrow (iii)$, we express $\mathcal{C}_Q^{D'}$ in its third form. Straightforward calculations give

$$\mathcal{C}_Q^{D'} \;=\; 1 - \frac{\big(1 - 2\,R_{D'}(G_Q)\big)^2}{1 - 2\,d_Q^{D'}} \;\leq\; 2R_{D'}(G_Q) \quad\Longleftrightarrow\quad R_{D'}(G_Q) \;\leq\; d_Q^{D'}\,.$$

■

Propositions 13 and 14 illustrate an interesting result: the $\mathcal{C}$-bound is optimal if and only if its value is lower than twice the Gibbs risk, the classical bound on the risk of the majority vote (see Proposition 10).

4.4.2 The $\mathcal{C}$-bound Can Be Arbitrarily Small, Even for Large Gibbs Risks

The next result shows that, when the number of voters tends to infinity (and the weight of each voter tends to zero), the variance of M_Q will tend to 0 provided that the average of the covariance of the outputs of all pairs of distinct voters is ≤ 0. In particular, the variance will always tend to 0 if the risk of the voters is pairwise independent. To quantify the independence between voters, we use the concept of covariance of a pair of voters (f_1, f_2):

$$\begin{aligned}\mathrm{Cov}^{D'}(f_1,f_2) &\stackrel{\text{def}}{=} \mathop{\mathbf{Cov}}_{(x,y)\sim D'}\Big(y\cdot f_1(x),\, y\cdot f_2(x)\Big)\\ &= \mathop{\mathbf{E}}_{(x,y)\sim D'} f_1(x)f_2(x) - \left(\mathop{\mathbf{E}}_{(x,y)\sim D'} f_1(x)\right)\left(\mathop{\mathbf{E}}_{(x,y)\sim D'} f_2(x)\right).\end{aligned}$$

Note that the covariance $\mathrm{Cov}^{D'}(f_1, f_2)$ is zero when f_1 and f_2 are independent (uncorrelated).

Proposition 15 *For any countable set of voters $\mathcal{H}$, any distribution Q on $\mathcal{H}$, and any distribution D' on $\mathcal{X}\times\{-1,1\}$, we have*

$$\mathrm{Var}(M_Q^{D'}) \;\leq\; \sum_{f\in\mathcal{H}} Q^2(f) \;+\; \sum_{f_1\in\mathcal{H}}\sum_{\substack{f_2\in\mathcal{H}:\\ f_2\neq f_1}} Q(f_1)Q(f_2)\cdot\mathrm{Cov}^{D'}(f_1,f_2)\,.$$

Proof By the definition of the margin (Definition 8), we rewrite $M_Q(x,y)$ as a sum of random variables:

$$\begin{aligned}&\mathop{\mathbf{Var}}_{(x,y)\sim D'}\Big(M_Q(x,y)\Big)\\ &\quad= \mathop{\mathbf{Var}}_{(x,y)\sim D'}\left(\sum_{f\in\mathcal{H}} Q(f)\cdot y\cdot f(x)\right)\\ &\quad= \sum_{f\in\mathcal{H}} Q^2(f)\mathop{\mathbf{Var}}_{(x,y)\sim D'}\Big(y\cdot f(x)\Big) + \sum_{f_1\in\mathcal{H}}\sum_{\substack{f_2\in\mathcal{H}:\\ f_2\neq f_1}} Q(f_1)Q(f_2)\mathop{\mathbf{Cov}}_{(x,y)\sim D'}\Big(y\cdot f_1(x), y\cdot f_2(x)\Big).\end{aligned}$$

The inequality is a consequence of the fact that $\forall f\in\mathcal{H}: \mathop{\mathbf{Var}}_{(x,y)\sim D'}\Big(y\cdot f(x)\Big)\leq 1$. ■

The key observation that comes out of this result is that $\sum_{f\in\mathcal{H}} Q^2(f)$ is usually much smaller than one. Consider, for example, the case where Q is uniform on $\mathcal{H}$ with $|\mathcal{H}| = n$. Then $\sum_{f\in\mathcal{H}} Q^2(f) = 1/n$. Moreover, if $\mathrm{Cov}^{D'}(f_1, f_2)\leq 0$ for each pair of distinct classifiers in $\mathcal{H}$, then $\mathrm{Var}(M_Q^{D'})\leq 1/n$. Hence, in these cases, we have that $\mathcal{C}_Q^{D'}\in\mathcal{O}(1/n)$ whenever $1-2\,R_{D'}(G_Q)$ and $1-2\,d_Q^{D'}$ are larger than some positive constants independent of n. Thus, even when $R_{D'}(G_Q)$ is large, we see that the $\mathcal{C}$-bound can be arbitrarily close to 0 as we increase the number of classifiers having non-positive pairwise covariance of their risk. More precisely, we have

Corollary 16 *Given n independent voters under a uniform distribution Q, we have*

$$R_{D'}(B_Q) \;\leq\; \mathcal{C}_Q^{D'} \;\leq\; \frac{1}{n\cdot\Big(1-2\,d_Q^{D'}\Big)} \;\leq\; \frac{1}{n\cdot\Big(1-2\,R_{D'}(G_Q)\Big)^2}\,.$$

Proof The first inequality directly comes from the $\mathcal{C}$-bound (Theorem 11). The second inequality is a consequence of Proposition 15, considering that in the case of a uniform distribution of independent voters, we have $\mathrm{Cov}^{D'}(f_1, f_2) = 0$, and then $\mathrm{Var}(M_Q^{D'}) \leq 1/n$. Applying this to the first form of the $\mathcal{C}$-bound, we obtain

$$\mathcal{C}_Q^{D'} \;=\; \frac{\mathrm{Var}(M_Q^{D'})}{\mu_2(M_Q^{D'})} \;=\; \frac{\mathrm{Var}(M_Q^{D'})}{1-2\,d_Q^{D'}} \;\leq\; \frac{\frac{1}{n}}{1-2\,d_Q^{D'}} \;=\; \frac{1}{n\cdot\left(1-2\,d_Q^{D'}\right)}\,.$$

To obtain the third inequality, we simply apply Equation (11), and we are done. ■

4.5 Empirical Study of The Predictive Power of the $\mathcal{C}$-bound

To further motivate the use of the $\mathcal{C}$-bound, we investigate how its empirical value relates to the risk of the majority vote by conducting two experiments. The first experiment shows that the $\mathcal{C}$-bound clearly outperforms the individual capacity of the other quantities of Theorem 11 in the task of predicting the risk of the majority vote. The second experiment shows that the $\mathcal{C}$-bound is a great stopping criterion for Boosting algorithms.

4.5.1 Comparison with Other Indicators

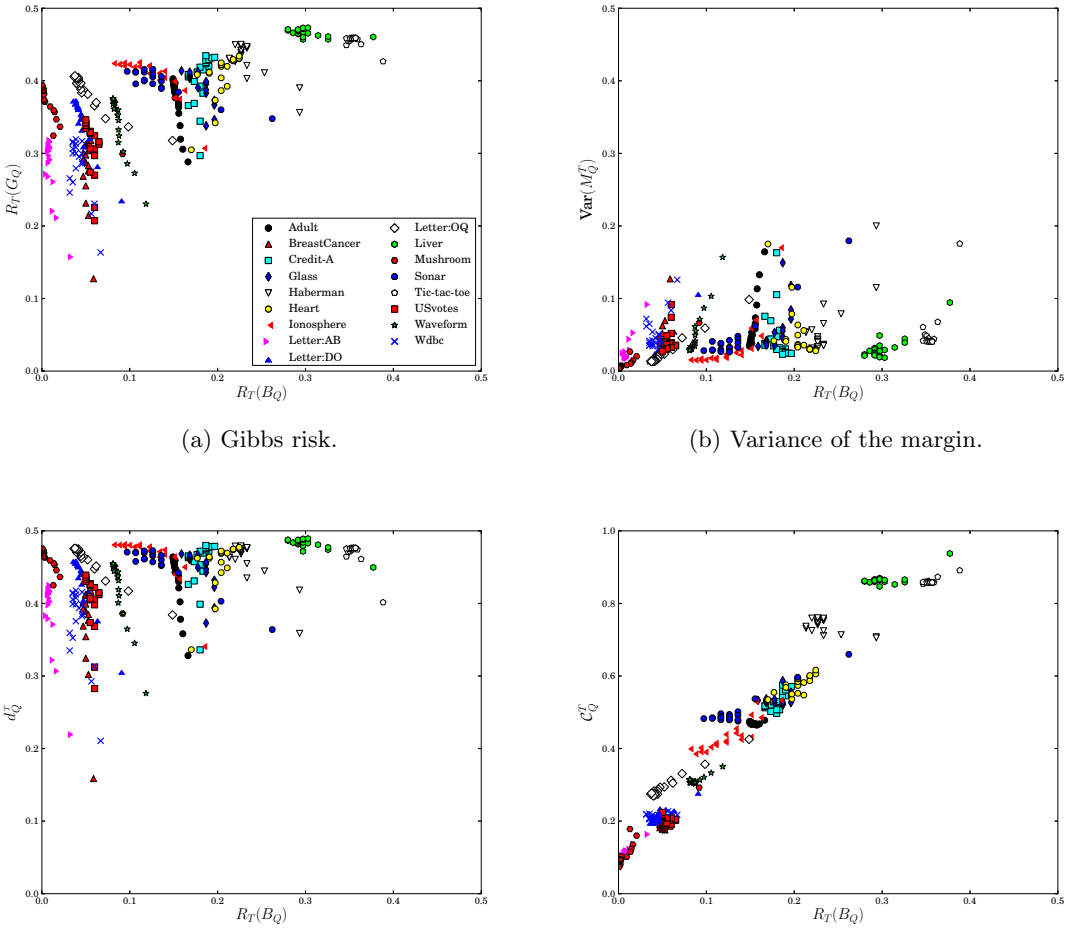
We study how $R_{D'}(G_Q)$, $\mathrm{Var}(M_Q^{D'})$, $d_Q^{D'}$ and $\mathcal{C}_Q^{D'}$ are respectively related to $R_{D'}(B_Q)$. Note that these four quantities appear in the first form or the third form of the $\mathcal{C}$-bound (Theorem 11). We omit here the moments $\mu_1(M_Q^{D'})$ and $\mu_2(M_Q^{D'})$ required by the second form of the $\mathcal{C}$-bound, as there is a linear relation between $\mu_1(M_Q^{D'})$ and $R_{D'}(G_Q)$, as well as between $\mu_2(M_Q^{D'})$ and $d_Q^{D'}$.

The results of Figure 3 are obtained with the AdaBoost algorithm of Schapire and Singer (1999), used with "decision stumps" as weak learners, on several UCI binary classification data sets (Blake and Merz, 1998). Each data set is split into two halves: a training set S and a testing set T. We run AdaBoost on set S for 100 rounds and compute the quantities $R_T(G_Q)$, $\mathrm{Var}(M_Q^T)$, d_Q^T and $\mathcal{C}_Q^T$ on set T at every 5 rounds of boosting. That is, we study 20 different majority vote classifiers per data set.

In Figure 3a, we see that we almost always have $R_T(B_Q) < R_T(G_Q)$. There is, however, no clear correlation between $R_T(B_Q)$ and $R_T(G_Q)$. We also see no clear correlation between $R_T(B_Q)$ and $\mathrm{Var}(M_Q^T)$ or between $R_T(B_Q)$ and d_Q^T in Figures 3b and 3c respectively, except that generally $R_T(B_Q) > \mathrm{Var}(M_Q^T)$ and $R_T(B_Q) < d_Q^T$. In contrast, Figure 3d shows a strong correlation between $\mathcal{C}_Q^T$ and $R_T(B_Q)$. Indeed, it is almost a linear relation! Therefore, the $\mathcal{C}$-bound seems well-suited to characterize the behavior of the Bayes risk, whereas each of the individual quantities contained in the $\mathcal{C}$-bound is insufficient to do so.

4.5.2 The $\mathcal{C}$-bound as a Stopping Criterion for Boosting

We now evaluate the accuracy of the empirical value of the $\mathcal{C}$-bound as a model selection tool. More specifically, we compare its ability to act as a stopping criterion for the AdaBoost algorithm.



(a) Gibbs risk.

(b) Variance of the margin.

(c) Expected disagreement.

(d) $\mathcal{C}$-bound.

Figure 3: $R_T(B_Q)$ versus $R_T(G_Q)$, $\mathrm{Var}(M_Q^T)$, d_Q^T and $\mathcal{C}_Q^T$ respectively.

We use the same version of the algorithm and the same data sets as in the previous experiment. However, for this experiment, each data set is split into a training set S of at most 400 examples and a testing set T containing the remaining examples. We run AdaBoost on set S for 1000 rounds. At each round, we compute the empirical $\mathcal{C}$-bound $\mathcal{C}_Q^S$ (on the training set). Afterwards, we select the majority vote classifier with the lowest value of $\mathcal{C}_Q^S$ and compute its Bayes risk $R_T(B_Q)$ (on the test set). We compare this stopping criterion with three other methods. For the first method, we compute the empirical Bayes risk $R_S(B_Q)$ at each round of boosting and, after that, we select the one having the lowest such risk.[7] The second method consists in performing 5-fold cross-validation and selecting the number of boosting rounds having the lowest cross-validation risk. Finally, the third method is to reserve 10% of S as a validation set, train AdaBoost on the remaining 90%,

7. When several iterations have the same value of $R_S(B_Q)$, we select the earlier one.

Data Set Information			Risk $R_T(B_Q)$ by Stopping Criterion *(and number of rounds performed)*								
Name	$\|S\|$	$\|T\|$	$\mathcal{C}$-bound $\mathcal{C}_Q^S$		Risk $R_S(B_Q)$		Validation Set		Cross-Validation		1000 rounds
Adult	400	11409	0.166	*(149)*	0.169	*(314)*	**0.165**	*(13)*	0.166	*(97)*	0.172
BreastCancer	341	342	0.050	*(127)*	0.047	*(48)*	**0.041**	*(57)*	0.047	*(108)*	0.058
Credit-A	326	327	0.187	*(346)*	0.199	*(854)*	**0.156**	*(9)*	0.174	*(47)*	0.199
Glass	107	107	0.252	*(72)*	**0.196**	*(299)*	0.346	*(6)*	0.290	*(35)*	**0.196**
Haberman	147	147	0.320	*(27)*	0.320	*(45)*	**0.279**	*(1)*	0.320	*(38)*	0.340
Heart	148	149	0.215	*(124)*	0.289	*(950)*	**0.181**	*(31)*	0.195	*(14)*	0.289
Ionosphere	175	176	**0.085**	*(210)*	0.120	*(56)*	0.142	*(2)*	0.114	*(67)*	**0.085**
Letter:AB	400	1155	**0.005**	*(42)*	0.014	*(17)*	0.061	*(2)*	**0.005**	*(60)*	0.010
Letter:DO	400	1158	**0.041**	*(179)*	**0.041**	*(44)*	0.143	*(1)*	0.044	*(83)*	0.043
Letter:OQ	400	1136	0.050	*(65)*	0.050	*(138)*	0.063	*(26)*	**0.044**	*(118)*	0.049
Liver	172	173	**0.289**	*(541)*	**0.289**	*(743)*	0.335	*(5)*	**0.289**	*(603)*	0.295
Mushroom	400	7724	**0.010**	*(612)*	0.024	*(38)*	0.079	*(6)*	0.024	*(51)*	**0.010**
Sonar	104	104	0.192	*(688)*	0.250	*(20)*	0.317	*(2)*	**0.163**	*(34)*	0.202
Tic-tac-toe	400	558	0.389	*(59)*	0.364	*(2)*	**0.358**	*(5)*	0.403	*(9)*	0.389
USvotes	217	218	0.032	*(11)*	0.041	*(598)*	0.032	*(16)*	**0.028**	*(1)*	0.046
Waveform	400	7600	**0.101**	*(145)*	0.102	*(178)*	0.106	*(13)*	0.103	*(22)*	0.115
Wdbc	284	285	0.049	*(40)*	0.060	*(19)*	0.091	*(2)*	**0.046**	*(10)*	0.060

Statistical Comparison Tests				
	$\mathcal{C}_Q^S$ vs $R_S(B_Q)$	$\mathcal{C}_Q^S$ vs Validation Set	$\mathcal{C}_Q^S$ vs Cross-Validation	$\mathcal{C}_Q^S$ vs 1000 rounds
Poisson binomial test	91%	86%	57%	90%
Sign test (p-value)	0.05	0.23	0.60	0.02

Table 1: Comparison of various stopping criteria over 1000 rounds of boosting. The Poisson binomial test gives the probability that $\mathcal{C}_Q^S$ is a better stopping criterion than every other approach. The sign test gives a p-value representing the probability that the null hypothesis is true (*i.e.*, the $\mathcal{C}_Q^S$ stopping criterion has the same performance as every other approach).

and keep the majority vote with the lowest Bayes risk on the validation set. Note that this last method differs from the others because AdaBoost sees 10% fewer examples during the learning process, but this is the price to pay for using a validation set.

Table 1 compares the Bayes risks on the test set $R_T(B_Q)$ of the majority vote classifiers selected by the different stopping criteria. We compute the probability of $\mathcal{C}$-bound being a better stopping criteria than every other methods with two statistical tests: the Poisson binomial test (Lacoste et al., 2012) and the sign test (Mendenhall, 1983). Both statistical tests suggest that the empirical $\mathcal{C}$-bound is a better model selection tool than the empirical Bayes risk (as usual in machine learning tasks, this method is prone to overfitting) and the validation set (although this method performs very well sometimes, it suffers from the small quantity of training examples on several tasks). The empirical $\mathcal{C}$-bound and the cross-validation methods obtain a similar accuracy. However, the cross-validation procedure needs more running time. We conclude that the empirical $\mathcal{C}$-bound is a surprisingly good stopping criterion for Boosting.

5. A PAC-Bayesian Story: From Zero to a PAC-Bayesian $\mathcal{C}$-bound

In this section, we present a PAC-Bayesian theory that allows one to estimate the $\mathcal{C}$-bound value $\mathcal{C}_Q^D$ from its empirical estimate $\mathcal{C}_Q^S$. From there, we derive bounds on the risk of the majority vote $R_D(B_Q)$ based on empirical observations. We first recall the classical PAC-Bayesian bound (here called the PAC-Bound 0) that bounds the true Gibbs risk by its empirical counterpart. We then present two different PAC-Bayesian bounds on the majority vote classifier (respectively called PAC-Bounds 1 and 2). A third bound, PAC-Bound 3, will be presented in Section 6. This analysis intends to be self-contained, and can act as an introduction to PAC-Bayesian theory.[8]

The first PAC-Bayesian theorem was proposed by McAllester (1999). Given a set of voters $\mathcal{H}$, a *prior* distribution P on $\mathcal{H}$ chosen before observing the data, and a *posterior* distribution Q on $\mathcal{H}$ chosen after observing a training set $S \sim D^m$ (Q is typically chosen by running a learning algorithm on S), PAC-Bayesian theorems give tight risk bounds for the Gibbs classifier G_Q. These bounds on $R_D(G_Q)$ usually rely on two quantities:

a) The empirical Gibbs risk $R_S(G_Q)$, that is computed on the m examples of S,

$$R_S(G_Q) \;=\; \frac{1}{m}\sum_{i=1}^{m} \mathop{\mathbf{E}}_{f\sim Q} \mathcal{L}_\ell(f(x_i), y_i)\,.$$

b) The Kullback-Leibler divergence between distributions Q and P, that measures "how far" the chosen posterior Q is from the prior P,

$$\mathrm{KL}(Q\|P) \;\stackrel{\text{def}}{=}\; \mathop{\mathbf{E}}_{f\sim Q} \ln \frac{Q(f)}{P(f)}\,. \tag{20}$$

Note that the obtained PAC-Bayesian bounds are uniformly valid for all possible posteriors Q.

In the following, we present a very general PAC-Bayesian theorem (Section 5.1), and we specialize it to obtain a bound on the Gibbs risk $R_D(G_Q)$ that is converted in a bound on the risk of the majority vote $R_D(B_Q)$ by the factor 2 of Proposition 10 (Section 5.2). Then, we define new losses that rely on a pair of voters (Section 5.3). These new losses allow us to extend the PAC-Bayesian theory to directly bound $R_D(B_Q)$ through the $\mathcal{C}$-bound (Sections 5.4 and 5.5). For each proposed bound, we explain the algorithmic procedure required to compute its value.

5.1 General PAC-Bayesian Theory for Real-Valued Losses

A key step of most PAC-Bayesian proofs is summarized by the following *Change of measure inequality* (Lemma 17).

We present here the same proof as in Seldin and Tishby (2010) and McAllester (2013). Note that the same result is derived from Fenchel's inequality in Banerjee (2006) and Donsker-Varadhan's variational formula for relative entropy in Seldin et al. (2012); Tolstikhin and Seldin (2013).

8. We also recommend the "practical prediction tutorial" of Langford (2005), that contains an insightful PAC-Bayesian introduction.

Lemma 17 (Change of measure inequality) *For any set $\mathcal{H}$, for any distributions P and Q on $\mathcal{H}$, and for any measurable function $\phi : \mathcal{H} \to \mathbb{R}$, we have*

$$\mathop{\mathbf{E}}_{f\sim Q} \phi(f) \;\leq\; \mathrm{KL}(Q\|P) + \ln\left(\mathop{\mathbf{E}}_{f\sim P} e^{\phi(f)}\right).$$

Proof The result is obtained by simple calculations, exploiting the definition of the KL-divergence given by Equation (20), and then Jensen's inequality (Lemma 47, in Appendix A) on concave function $\ln(\cdot)$:

$$\begin{aligned}
\mathop{\mathbf{E}}_{f\sim Q} \phi(f) &= \mathop{\mathbf{E}}_{f\sim Q} \ln e^{\phi(f)} = \mathop{\mathbf{E}}_{f\sim Q} \ln\left(\frac{Q(f)}{P(f)}\cdot\frac{P(f)}{Q(f)}\cdot e^{\phi(f)}\right)\\
&= \mathrm{KL}(Q\|P) + \mathop{\mathbf{E}}_{f\sim Q} \ln\left(\frac{P(f)}{Q(f)}\cdot e^{\phi(f)}\right)\\
&\leq \mathrm{KL}(Q\|P) + \ln\left(\mathop{\mathbf{E}}_{f\sim Q} \frac{P(f)}{Q(f)}\cdot e^{\phi(f)}\right) \qquad \text{(Jensen's inequality)}\\
&\leq \mathrm{KL}(Q\|P) + \ln\left(\mathop{\mathbf{E}}_{f\sim P} e^{\phi(f)}\right).
\end{aligned}$$

Note that the last inequality becomes an equality if Q and P share the same support. ■

Let us now present a general PAC-Bayesian theorem which bounds the expectation of any real-valued loss function $\mathcal{L} : \overline{\mathcal{Y}} \times \mathcal{Y} \to [0,1]$. This theorem is slightly more general than the PAC-Bayesian theorem of Germain et al. (2009, Theorem 2.1), that is specialized to the expected linear loss, and therefore gives rise to a bound of the "generalized" Gibbs risk of Definition 5. A similar result is presented in Tolstikhin and Seldin (2013, Lemma 1).

Theorem 18 (General PAC-Bayesian theorem for real-valued losses) *For any distribution D on $\mathcal{X}\times\mathcal{Y}$, for any set $\mathcal{H}$ of voters $\mathcal{X}\to\overline{\mathcal{Y}}$, for any loss $\mathcal{L} : \overline{\mathcal{Y}}\times\mathcal{Y}\to[0,1]$, for any prior distribution P on $\mathcal{H}$, for any $\delta\in(0,1]$, for any $m' > 0$, and for any convex function $\mathcal{D} : [0,1]\times[0,1]\to\mathbb{R}$, we have*

$$\Pr_{S\sim D^m}\left(\begin{array}{l}\text{For all posteriors } Q \text{ on } \mathcal{H}:\\ \mathcal{D}\big(\mathop{\mathbf{E}}_{f\sim Q}\mathbb{E}^{\mathcal{L}}_S(f), \mathop{\mathbf{E}}_{f\sim Q}\mathbb{E}^{\mathcal{L}}_D(f)\big) \leq \dfrac{1}{m'}\left[\mathrm{KL}(Q\|P)+\ln\left(\dfrac{1}{\delta}\mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{f\sim P} e^{\,m'\cdot\mathcal{D}(\mathbb{E}^{\mathcal{L}}_S(f),\mathbb{E}^{\mathcal{L}}_D(f))}\right)\right]\end{array}\right) \geq 1-\delta\,,$$

where $\mathrm{KL}(Q\|P)$ is the Kullback-Leibler divergence between Q and P of Equation (20).

Most of the time, this theorem is used with $m' = m$, the size of the training set. However, as pointed out by Lever et al. (2010), m' does not have to be so. One can easily show that different values of m' affect the relative weighting between the terms $\mathrm{KL}(Q\|P)$ and $\ln\left(\frac{1}{\delta}\mathbf{E}_{S\sim D^m}\mathbf{E}_{f\sim P}e^{\,m'\cdot\mathcal{D}(\mathbb{E}^{\mathcal{L}}_S(f),\mathbb{E}^{\mathcal{L}}_D(f))}\right)$ in the bound. Hence, especially in situations where these two terms have very different values, a "good" choice for the value of m' can tighten the bound.

Proof Note that $\mathop{\mathbf{E}}_{f\sim P} e^{m'\cdot\mathcal{D}(\mathbb{E}^{\mathcal{L}}_S(f),\mathbb{E}^{\mathcal{L}}_D(f))}$ is a non-negative random variable. By Markov's inequality (Lemma 46, in Appendix A), we have

$$\Pr_{S\sim D^m}\left(\mathop{\mathbf{E}}_{f\sim P} e^{m'\cdot\mathcal{D}(\mathbb{E}^{\mathcal{L}}_S(f),\mathbb{E}^{\mathcal{L}}_D(f))} \leq \frac{1}{\delta}\mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{f\sim P} e^{m'\cdot\mathcal{D}(\mathbb{E}^{\mathcal{L}}_S(f),\mathbb{E}^{\mathcal{L}}_D(f))}\right) \geq 1-\delta\,.$$

Hence, by taking the logarithm on each side of the innermost inequality, we obtain

$$\Pr_{S\sim D^m}\left(\ln\left[\mathop{\mathbf{E}}_{f\sim P} e^{m'\cdot\mathcal{D}(\mathbb{E}_S^{\mathcal{L}}(f),\mathbb{E}_D^{\mathcal{L}}(f))}\right] \le \ln\left[\frac{1}{\delta}\mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{f\sim P} e^{m'\cdot\mathcal{D}(\mathbb{E}_S^{\mathcal{L}}(f),\mathbb{E}_D^{\mathcal{L}}(f))}\right]\right) \ge 1-\delta\,.$$

We apply the change of measure inequality (Lemma 17) on the left side of innermost inequality, with $\phi(f) = m'\cdot\mathcal{D}(\mathbb{E}_S^{\mathcal{L}}(f),\mathbb{E}_D^{\mathcal{L}}(f))$. We then use Jensen's inequality (Lemma 47, in Appendix A), exploiting the convexity of $\mathcal{D}$:

$$\begin{aligned}\forall\, Q \text{ on } \mathcal{H}: \quad \ln\left[\mathop{\mathbf{E}}_{f\sim P} e^{m'\cdot\mathcal{D}(\mathbb{E}_S^{\mathcal{L}}(f),\mathbb{E}_D^{\mathcal{L}}(f))}\right] &\ge m'\cdot\mathop{\mathbf{E}}_{f\sim Q}\mathcal{D}(\mathbb{E}_S^{\mathcal{L}}(f),\mathbb{E}_D^{\mathcal{L}}(f)) - \mathrm{KL}(Q\|P)\\ &\ge m'\cdot\mathcal{D}(\mathop{\mathbf{E}}_{f\sim Q}\mathbb{E}_S^{\mathcal{L}}(f),\mathop{\mathbf{E}}_{f\sim Q}\mathbb{E}_D^{\mathcal{L}}(f)) - \mathrm{KL}(Q\|P)\,.\end{aligned}$$

We therefore have

$$\Pr_{S\sim D^m}\left(\begin{array}{l}\text{For all posteriors } Q:\\ m'\cdot\mathcal{D}(\mathop{\mathbf{E}}_{f\sim Q}\mathbb{E}_S^{\mathcal{L}}(f),\mathop{\mathbf{E}}_{f\sim Q}\mathbb{E}_D^{\mathcal{L}}(f)) - \mathrm{KL}(Q\|P) \le \ln\left[\frac{1}{\delta}\mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{f\sim P} e^{m'\cdot\mathcal{D}(\mathbb{E}_S^{\mathcal{L}}(f),\mathbb{E}_D^{\mathcal{L}}(f))}\right]\end{array}\right) \ge 1-\delta\,.$$

The result then follows from easy calculations. ■

As shown in Germain et al. (2009), the general PAC-Bayesian theorem can be used to recover many common variants of the PAC-Bayesian theorem, simply by selecting a well-suited function $\mathcal{D}$. Among these, we obtain a similar bound as the one proposed by Langford and Seeger (2001); Seeger (2002); Langford (2005) by using the Kullback-Leibler divergence between the Bernoulli distributions with probability of success q and probability of success p:

$$\mathrm{kl}\big(q\,\|\,p\big) \stackrel{\text{def}}{=} q\ln\frac{q}{p} + (1-q)\ln\frac{1-q}{1-p}\,. \tag{21}$$

Note that $\mathrm{kl}\big(q\,\|\,p\big)$ is a shorthand notation for $\mathrm{KL}(Q\|P)$ of Equation (20), with $Q = (q, 1{-}q)$ and $P = (p, 1{-}p)$. Corollary 50 (in Appendix A) shows that $\mathrm{kl}\big(q\,\|\,p\big)$ is a convex function.

In order to apply Theorem 18 with $\mathcal{D}(q,p) = \mathrm{kl}(q\|p)$ and $m' = m$, we need the next lemma.

Lemma 19 *For any distribution D on $\mathcal{X}\times\mathcal{Y}$, for any voter $f : \mathcal{X}\to\overline{\mathcal{Y}}$, for any loss $\mathcal{L} : \overline{\mathcal{Y}}\times\mathcal{Y}\to[0,1]$, and any positive integer m, we have*

$$\mathop{\mathbf{E}}_{S\sim D^m}\exp\left[m\cdot\mathrm{kl}\Big(\mathbb{E}_S^{\mathcal{L}}(f)\,\|\,\mathbb{E}_D^{\mathcal{L}}(f)\Big)\right] \;\le\; \xi(m)\,,$$

where

$$\xi(m) \stackrel{\text{def}}{=} \sum_{k=0}^{m}\binom{m}{k}\left(\frac{k}{m}\right)^k\left(1-\frac{k}{m}\right)^{m-k}. \tag{22}$$

Moreover, $\sqrt{m} \le \xi(m) \le 2\sqrt{m}$.

Proof Let us introduce a random variable X_f that follows a binomial distribution of m trials with a probability of success $\mathbb{E}_D^{\mathcal{L}}(f)$. Hence, $X_f \sim B(m, \mathbb{E}_D^{\mathcal{L}}(f))$.

As $e^{m\cdot \mathrm{kl}\left(\cdot \,\|\, \mathbb{E}_D^{\mathcal{L}}(f)\right)}$ is a convex function, Lemma 51 (due to Maurer, 2004, and provided in Appendix A), shows that

$$\mathop{\mathbf{E}}_{S\sim D^m} \exp\left[m\cdot \mathrm{kl}\Big(\mathbb{E}_S^{\mathcal{L}}(f) \,\|\, \mathbb{E}_D^{\mathcal{L}}(f)\Big)\right] \;\le\; \mathop{\mathbf{E}}_{X_f\sim B(m,\mathbb{E}_D^{\mathcal{L}}(f))} \exp\left[m\cdot \mathrm{kl}\Big(\tfrac{1}{m}X_f \,\|\, \mathbb{E}_D^{\mathcal{L}}(f)\Big)\right].$$

We then have

$$\begin{aligned}
&\mathop{\mathbf{E}}_{X_f\sim B(m,\mathbb{E}_D^{\mathcal{L}}(f))} e^{m\mathrm{kl}(\frac{1}{m}X_f\|\mathbb{E}_D^{\mathcal{L}}(f))} \\
&\quad= \mathop{\mathbf{E}}_{X_f\sim B(m,\mathbb{E}_D^{\mathcal{L}}(f))} \left(\frac{\frac{1}{m}X_f}{\mathbb{E}_D^{\mathcal{L}}(f)}\right)^{X_f}\left(\frac{1-\frac{1}{m}X_f}{1-\mathbb{E}_D^{\mathcal{L}}(f)}\right)^{m-X_f} \\
&\quad= \sum_{k=0}^{m} \Pr_{X_f\sim B(m,\mathbb{E}_D^{\mathcal{L}}(f))}\Big(X_f=k\Big)\cdot\left(\frac{\frac{k}{m}}{\mathbb{E}_D^{\mathcal{L}}(f)}\right)^{k}\left(\frac{1-\frac{k}{m}}{1-\mathbb{E}_D^{\mathcal{L}}(f)}\right)^{m-k} \\
&\quad= \sum_{k=0}^{m}\binom{m}{k}\Big(\mathbb{E}_D^{\mathcal{L}}(f)\Big)^{k}\Big(1-\mathbb{E}_D^{\mathcal{L}}(f)\Big)^{m-k}\cdot\left(\frac{\frac{k}{m}}{\mathbb{E}_D^{\mathcal{L}}(f)}\right)^{k}\left(\frac{1-\frac{k}{m}}{1-\mathbb{E}_D^{\mathcal{L}}(f)}\right)^{m-k} \\
&\quad= \sum_{k=0}^{m}\binom{m}{k}\left(\frac{k}{m}\right)^{k}\left(1-\frac{k}{m}\right)^{m-k} \;=\; \xi(m)\,.
\end{aligned}$$

Maurer (2004) shows that $\xi(m)\le 2\sqrt{m}$ for $m\ge 8$, and $\xi(m)\ge\sqrt{m}$ for $m\ge 2$. However, the cases for $m\in\{1,2,3,4,5,6,7\}$ are easy to verify computationally. ■

Theorem 20 below specializes the general PAC-Bayesian theorem to $\mathcal{D}(q,p)=\mathrm{kl}(q\|p)$, but still applies to any real-valued loss functions. This theorem can be seen as an intermediate step to obtain Corollary 21 of the next section, which uses the linear loss to bound the Gibbs risk. However, Theorem 20 below is reused afterwards in Section 5.3 to derive PAC-Bayesian theorems for other loss functions.

Theorem 20 *For any distribution D on $\mathcal{X}\times\mathcal{Y}$, for any set $\mathcal{H}$ of voters $\mathcal{X}\to\overline{\mathcal{Y}}$, for any loss $\mathcal{L}:\overline{\mathcal{Y}}\times\mathcal{Y}\to[0,1]$, for any prior distribution P on $\mathcal{H}$, for any $\delta\in(0,1]$, we have*

$$\Pr_{S\sim D^m}\left(\begin{array}{l}\text{For all posteriors } Q \text{ on } \mathcal{H}:\\ \mathrm{kl}\Big(\mathop{\mathbf{E}}_{f\sim Q}\mathbb{E}_S^{\mathcal{L}}(f)\,\Big\|\,\mathop{\mathbf{E}}_{f\sim Q}\mathbb{E}_D^{\mathcal{L}}(f)\Big)\;\le\;\dfrac{1}{m}\left[\mathrm{KL}(Q\|P)+\ln\dfrac{\xi(m)}{\delta}\right]\end{array}\right)\ge 1-\delta\,.$$

Proof By Theorem 18, with $\mathcal{D}(q,p)=\mathrm{kl}(q\|p)$ and $m'=m$, we have

$$\Pr_{S\sim D^m}\left(\begin{array}{l}\forall\, Q \text{ on } \mathcal{H}:\\ \mathrm{kl}(\mathop{\mathbf{E}}_{f\sim Q}\mathbb{E}_S^{\mathcal{L}}(f)\,\|\,\mathop{\mathbf{E}}_{f\sim Q}\mathbb{E}_D^{\mathcal{L}}(f))\le\dfrac{1}{m}\left[\mathrm{KL}(Q\|P)+\ln\left(\dfrac{1}{\delta}\mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{f\sim P}e^{m\cdot\mathrm{kl}(\mathbb{E}_S^{\mathcal{L}}(f)\,\|\,\mathbb{E}_D^{\mathcal{L}}(f))}\right)\right]\end{array}\right)\ge 1-\delta\,.$$

As the prior P is independent of S, we can swap the two expectations in $\mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{f\sim P}e^{m\cdot\mathrm{kl}(\cdot\|\cdot)}$. This observation, together with Lemma 19, gives

$$\mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{f\sim P}e^{m\cdot\mathrm{kl}(\mathbb{E}_S^{\mathcal{L}}(f)\,\|\,\mathbb{E}_D^{\mathcal{L}}(f))} \;=\; \mathop{\mathbf{E}}_{f\sim P}\mathop{\mathbf{E}}_{S\sim D^m}e^{m\cdot\mathrm{kl}(\mathbb{E}_S^{\mathcal{L}}(f)\,\|\,\mathbb{E}_D^{\mathcal{L}}(f))} \;\le\; \mathop{\mathbf{E}}_{f\sim P}\xi(m) \;=\; \xi(m)\,.$$

■

5.2 PAC-Bayesian Theory for the Gibbs Classifier

This section presents two classical PAC-Bayesian results that bound the risk of the Gibbs classifier. One of these bounds is used to express a first PAC-Bayesian bound on the risk of the majority vote classifier. Then, we explain how to compute the empirical value of this bound by a root-finding method.

5.2.1 PAC-Bayesian Theorems for the Gibbs Risk

We interpret the two following results as straightforward corollaries of Theorem 20. Indeed, from Definition 5, the expected linear loss of a Gibbs classifier G_Q on a distribution D' *is* $R_{D'}(G_Q)$. These two Corollaries are very similar to well-known PAC-Bayesian theorems. At first, Corollary 21 is similar to the PAC-Bayesian theorem of Langford and Seeger (2001); Seeger (2002); Langford (2005), with the exception that $\ln \frac{m+1}{\delta}$ is replaced by $\ln \frac{\xi(m)}{\delta}$. Since $\xi(m) \leq 2\sqrt{m} \leq m+1$, this result gives slightly better bounds. Similarly, Corollary 22 provides a slight improvement of the PAC-Bayesian bound of McAllester (1999, 2003a).

Corollary 21 (Langford and Seeger, 2001; Seeger, 2002; Langford, 2005) *For any distribution D on $\mathcal{X}\times\{-1,1\}$, for any set $\mathcal{H}$ of voters $\mathcal{X}\to[-1,1]$, for any prior distribution P on $\mathcal{H}$, and any $\delta\in(0,1]$, we have*

$$\Pr_{S\sim D^m}\left(\begin{array}{l}\textit{For all posteriors } Q \textit{ on } \mathcal{H}:\\ \mathrm{kl}\big(R_S(G_Q)\big\|R_D(G_Q)\big) \;\leq\; \dfrac{1}{m}\left[\mathrm{KL}(Q\|P)+\ln\dfrac{\xi(m)}{\delta}\right]\end{array}\right)\geq 1-\delta\,.$$

Proof The result is directly obtained from Theorem 20 using the linear loss $\mathcal{L}=\mathcal{L}_\ell$ to recover the Gibbs risk of Definition 5. ■

Corollary 22 (McAllester, 1999, 2003a) *For any distribution D on $\mathcal{X}\times\{-1,1\}$, for any set $\mathcal{H}$ of voters $\mathcal{X}\to[-1,1]$, for any prior distribution P on $\mathcal{H}$, and any $\delta\in(0,1]$, we have*

$$\Pr_{S\sim D^m}\left(\begin{array}{l}\textit{For all posteriors } Q \textit{ on } \mathcal{H}:\\ R_D(G_Q) \;\leq\; R_S(G_Q)+\sqrt{\dfrac{1}{2m}\left[\mathrm{KL}(Q\|P)+\ln\dfrac{\xi(m)}{\delta}\right]}\end{array}\right)\geq 1-\delta\,.$$

Proof The result is obtained from Corollary 21 together with Pinsker's inequality

$$2(q-p)^2 \;\leq\; \mathrm{kl}(q\|p)\,.$$

We then have

$$\Pr_{S\sim D^m}\left(\begin{array}{l}\text{For all posteriors } Q \text{ on } \mathcal{H}:\\ 2\cdot\Big(R_S(G_Q)-R_D(G_Q)\Big)^2 \;\leq\; \dfrac{1}{m}\left[\mathrm{KL}(Q\|P)+\ln\dfrac{\xi(m)}{\delta}\right]\end{array}\right)\geq 1-\delta\,.$$

The result is obtained by isolating $R_D(G_Q)$ in the inequality, omitting the lower bound of $R_D(G_Q)$. Recall that the probability is "$\geq 1-\delta$", hence if we omit an event, the probability may just increase, continuing to be greater than $1-\delta$. ■

5.2.2 A First Bound for the Risk of the Majority Vote

Let assume that the Gibbs risk $R_D(G_Q)$ of a classifier is lower than or equal to $\frac{1}{2}$. Given an empirical Gibbs risk $R_S(G_Q)$ computed on a training set of m examples, the Kullback-Leibler divergence $\mathrm{KL}(Q\|P)$, and a confidence parameter δ, Corollary 21 says that the Gibbs risk $R_D(G_Q)$ is included (with confidence $1-\delta$) in the continuous set $\mathcal{R}^{\delta}_{Q,S}$ defined as

$$\mathcal{R}^{\delta}_{Q,S} \stackrel{\text{def}}{=} \left\{ r \,:\, \mathrm{kl}\big(R_S(G_Q) \,\big\|\, r\big) \,\leq\, \frac{1}{m}\left[\,\mathrm{KL}(Q\|P) + \ln\frac{\xi(m)}{\delta}\right] \quad \text{and} \quad r \leq \tfrac{1}{2} \right\}. \tag{23}$$

Thus, an upper bound on $R_D(G_Q)$ is obtained by seeking the maximum value of $\mathcal{R}^{\delta}_{Q,S}$. As explained by Proposition 10, we need to multiply the obtained value by a factor 2 to have an upper bound on $R_D(B_Q)$. This methodology is summarized by PAC-Bound 0.

Note that PAC-Bound 0 is also valid when $R_D(G_Q)$ is greater than $\frac{1}{2}$, because in this case, $2 \cdot \sup \mathcal{R}^{\delta}_{Q,S} = 1$ (with confidence at least $1-\delta$), which is a trivial upper bound of $R_D(B_Q)$.

PAC-Bound 0 *For any distribution D on $\mathcal{X}\times\{-1,1\}$, for any set $\mathcal{H}$ of voters $\mathcal{X} \to [-1,1]$, for any prior distribution P on $\mathcal{H}$, and any $\delta \in (0,1]$, we have*

$$\Pr_{S\sim D^m}\left(\forall\, Q \text{ on } \mathcal{H} \,:\, R_D(B_Q) \;\leq\; 2\cdot\sup\mathcal{R}^{\delta}_{Q,S}\right) \geq\; 1-\delta\,.$$

Proof If $\sup \mathcal{R}^{\delta}_{Q,S} = \frac{1}{2}$, the bound is trivially valid because $R_D(B_Q) \leq 1$. Otherwise, the bound is a direct consequence of Proposition 10 and Corollary 21. ■

As we see, the proposed bound cannot be obtained by a closed-form expression. Thus, we need to use a strategy as the one suggested in the following.

5.2.3 Computation of PAC-Bound 0

One can compute the value $r = \sup \mathcal{R}^{\delta}_{Q,S}$ of PAC-Bound 0 by solving

$$\mathrm{kl}\big(R_S(G_Q) \,\big\|\, r\big) \;=\; \tfrac{1}{m}\big[\mathrm{KL}(Q\|P) + \ln\tfrac{\xi(m)}{\delta}\big]\,, \qquad \text{with } R_S(G_Q) \leq r \leq \tfrac{1}{2}\,,$$

by a root-finding method. This turns out to be an easy task since the left-hand side of the equality is a convex function of r and the right-hand side is a constant value. Note that solving the same equation with the constraint $r \leq R_S(G_Q)$ gives a lower bound of $R_D(G_Q)$, but not a lower bound on $R_D(B_Q)$. Figure 4 shows an application example of PAC-Bound 0.

5.3 Joint Error, Joint Success, and Paired-voters

We now introduce a few notions that are necessary to obtain new PAC-Bayesian theorems for the $\mathcal{C}$-bound in Sections 5.4 and 5.5.

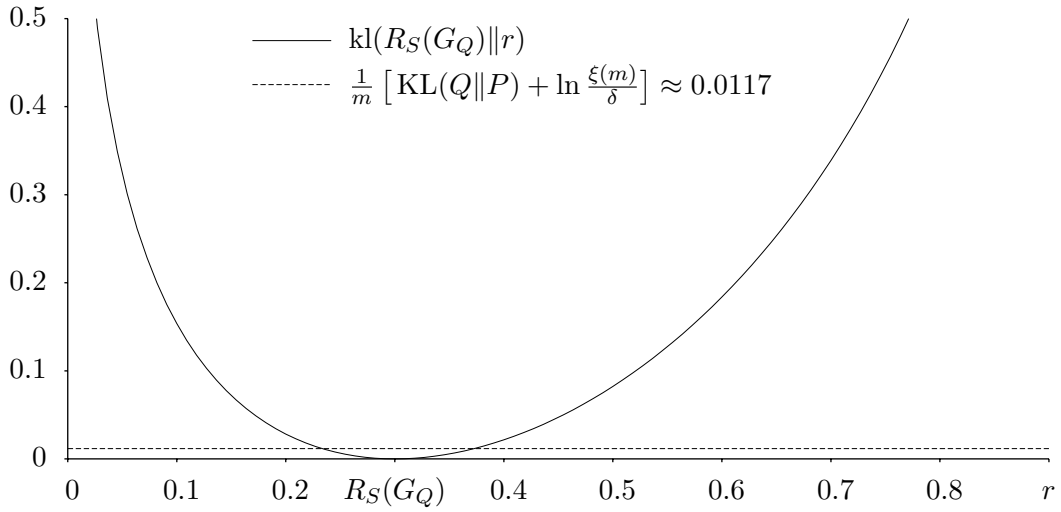


Figure 4: Example of application of PAC-Bound 0. We suppose that $\mathrm{KL}(Q\|P) = 5$, $m = 1000$ and $\delta = 0.05$. If we observe an empirical Gibbs risk $R_S(G_Q) = 0.30$, then $R_D(G_Q) \in \mathcal{R}^{\delta}_{Q,S} \approx [0.233, 0.373]$ with a confidence of 95%. On the figure, the intersections between the two curves correspond to the limits of the interval $\mathcal{R}^{\delta}_{Q,S}$. Then, with these values, PAC-bound 0 gives $R_D(B_Q) \lesssim 2 \cdot 0.373 = 0.746$.

5.3.1 The Joint Error and the Joint Success

We have already defined the expected disagreement $d_Q^{D'}$ of a distribution Q of voters (Definition 7). In the case of binary voters, the expected disagreement corresponds to

$$d_Q^{D'} \quad = \quad \mathop{\mathbf{E}}_{h_1\sim Q}\ \mathop{\mathbf{E}}_{h_2\sim Q}\ \left(\mathop{\mathbf{E}}_{(x,y)\sim D'} I(h_1(x) \neq h_2(x))\right).$$

Let us now define two closely related notions, the expected joint success $s_Q^{D'}$ and the expected joint error $e_Q^{D'}$. In the case of binary voters, these two concepts are expressed naturally by

$$\begin{aligned} e_Q^{D'} &= \mathop{\mathbf{E}}_{h_1\sim Q}\ \mathop{\mathbf{E}}_{h_2\sim Q}\ \left(\mathop{\mathbf{E}}_{(x,y)\sim D'} I(h_1(x) \neq y) I(h_2(x) \neq y)\right), \\ s_Q^{D'} &= \mathop{\mathbf{E}}_{h_1\sim Q}\ \mathop{\mathbf{E}}_{h_2\sim Q}\ \left(\mathop{\mathbf{E}}_{(x,y)\sim D'} I(h_1(x) = y) I(h_2(x) = y)\right). \end{aligned}$$

Let us now extend in the usual way these equations to the case of real-valued voters.

Definition 23 For any probability distribution Q on a set of voters, we define the *expected joint error* $e_Q^{D'}$ relative to D' and the *expected joint success* $s_Q^{D'}$ relative to D' as

$$\begin{aligned} e_Q^{D'} &\stackrel{\text{def}}{=} \mathop{\mathbf{E}}_{f_1\sim Q}\ \mathop{\mathbf{E}}_{f_2\sim Q}\ \left(\mathop{\mathbf{E}}_{(x,y)\sim D'} \mathcal{L}_\ell(f_1(x), y) \cdot \mathcal{L}_\ell(f_2(x), y)\right), \\ s_Q^{D'} &\stackrel{\text{def}}{=} \mathop{\mathbf{E}}_{f_1\sim Q}\ \mathop{\mathbf{E}}_{f_2\sim Q}\ \left(\mathop{\mathbf{E}}_{(x,y)\sim D'} \Big[1 - \mathcal{L}_\ell(f_1(x), y)\Big] \cdot \Big[1 - \mathcal{L}_\ell(f_2(x), y)\Big]\right). \end{aligned}$$

From the definitions of the linear loss (Definition 2) and the margin (Definition 8), we can easily see that

$$\begin{aligned} e_Q^{D'} &= \mathop{\mathbf{E}}_{(x,y)\sim D'} \left(\frac{1-M_Q(x,y)}{2}\right)^2 = \frac{1}{4}\Big(1-2\cdot\mu_1(M_Q^{D'})+\mu_2(M_Q^{D'})\Big)\,, \\ s_Q^{D'} &= \mathop{\mathbf{E}}_{(x,y)\sim D'} \left(\frac{1+M_Q(x,y)}{2}\right)^2 = \frac{1}{4}\Big(1+2\cdot\mu_1(M_Q^{D'})+\mu_2(M_Q^{D'})\Big)\,. \end{aligned}$$

Remembering from Equation (9) that $d_Q^{D'} = \frac{1}{2}\Big(1-\mu_2(M_Q^{D'})\Big)$, we can conclude that $e_Q^{D'}$, $s_Q^{D'}$ and $d_Q^{D'}$ always sum to one:[9]

$$e_Q^{D'} + s_Q^{D'} + d_Q^{D'} = 1\,.$$

We can now rewrite the first moment of the margin and the Gibbs risk as

$$\begin{aligned} \mu_1(M_Q^{D'}) &= s_Q^{D'} - e_Q^{D'} = 1-(2e_Q^{D'}+d_Q^{D'})\,, \\ R_{D'}(G_Q) &= \tfrac{1}{2}(1-s_Q^{D'}+e_Q^{D'}) = \tfrac{1}{2}(2e_Q^{D'}+d_Q^{D'})\,. \end{aligned} \tag{24}$$

Therefore, the third form of $\mathcal{C}$-bound of Theorem 11 can be rewritten as

$$\mathcal{C}_Q^{D'} = 1-\frac{\Big(1-(2e_Q^{D'}+d_Q^{D'})\Big)^2}{1-2d_Q^{D'}}\,. \tag{25}$$

5.3.2 Paired-Voters and Their Losses

This first generalization of the PAC-Bayesian theorem allows us to bound *separately* either d_Q^D, e_Q^D or s_Q^D, and therefore to bound $\mathcal{C}_Q^D$. To prove this result, we need to define a new kind of voter that we call a paired-voter.

Definition 24 Given two voters $f_i : \mathcal{X} \to [-1,1]$ and $f_j : \mathcal{X} \to [-1,1]$, the *paired-voter* $f_{ij} : \mathcal{X} \to [-1,1]^2$ outputs a tuple:

$$f_{ij}(x) \stackrel{\text{def}}{=} \langle\, f_i(x), f_j(x)\,\rangle\,.$$

Given a set of voters $\mathcal{H}$ weighted by a distribution Q on $\mathcal{H}$, we define a set of paired-voters $\mathcal{H}^2$ weighted by a distribution Q^2 as

$$\mathcal{H}^2 \stackrel{\text{def}}{=} \{f_{ij} \,:\, f_i, f_j \in \mathcal{H}\}\,, \quad \text{and} \quad Q^2(f_{ij}) \stackrel{\text{def}}{=} Q(f_i)\cdot Q(f_j)\,. \tag{26}$$

We now present three losses for paired-voters. Remember that a loss function has the form $\overline{\mathcal{Y}}\times\mathcal{Y} \to [0,1]$, where $\overline{\mathcal{Y}}$ is the voter's output space. As a paired-voter output is a

9. This is fairly intuitive in the case of binary voters. Indeed, given any example (x,y) and any two binary voters h_1, h_2, we have either: both voters misclassify the example – *i.e.*, $h_1(x) = h_2(x) \neq y$ –, both voters correctly classify the example – *i.e.*, $h_1(x) = h_2(x) = y$ –, or both voters disagree – *i.e.*, $h_1(x) \neq h_2(x)$.

tuple, our new loss functions map $[-1,1]^2 \times \{-1,1\}$ to $[0,1]$. Thus,

$$\begin{aligned}
\mathcal{L}_e\big(f_{ij}(x),\, y\big) &\stackrel{\text{def}}{=} \mathcal{L}_\ell(f_i(x),y)\cdot\mathcal{L}_\ell(f_j(x),y)\,,\\
\mathcal{L}_s\big(f_{ij}(x),\, y\big) &\stackrel{\text{def}}{=} \Big[1-\mathcal{L}_\ell(f_i(x),y)\Big]\cdot\Big[1-\mathcal{L}_\ell(f_j(x),y)\Big]\,,\\
\mathcal{L}_d\big(f_{ij}(x),\, y\big) &\stackrel{\text{def}}{=} \mathcal{L}_\ell(f_i(x)\cdot f_j(x)\,,\,1)\,. \qquad (27)
\end{aligned}$$

The key observation to understand the next theorems is that the expected losses of paired-voters $\mathcal{H}^2$ defined by Equation (26) allow one to recover the values of $e_Q^{D'}$, $s_Q^{D'}$ and $d_Q^{D'}$. Indeed, it directly follows from Definitions 3, 7 and 23, that

$$e_Q^{D'} = \mathop{\mathbf{E}}_{f_{ij}\sim Q^2}\mathbb{E}_{D'}^{\mathcal{L}_e}\Big(f_{ij}\Big)\,;\quad s_Q^{D'} = \mathop{\mathbf{E}}_{f_{ij}\sim Q^2}\mathbb{E}_{D'}^{\mathcal{L}_s}\Big(f_{ij}\Big)\,;\quad d_Q^{D'} = \mathop{\mathbf{E}}_{f_{ij}\sim Q^2}\mathbb{E}_{D'}^{\mathcal{L}_d}\Big(f_{ij}\Big)\,. \qquad (28)$$

5.4 PAC-Bayesian Theory For Losses of Paired-voters

As explained in Section 5.2, classical PAC-Bayesian theorems, like Corollaries 21 and 22, provide an upper bound on $R_D(G_Q)$ that holds uniformly for all posteriors Q. A bound on $R_D(B_Q)$ is typically obtained by multiplying the former bound by the usual factor of 2, as in PAC-Bound 0.

In this subsection, we present a first bound of $R_D(B_Q)$ relying on the $\mathcal{C}$-bound of Theorem 11. A uniform bound on $\mathcal{C}_Q^D$ is obtained using the third form of the $\mathcal{C}$-bound, through a bound on the Gibbs risk $R_D(G_Q)$ and another bound on the disagreement d_Q^D. The desired bound on $R_D(G_Q)$ is obtained by Corollary 21 as in PAC-Bound 0. To obtain a bound on d_Q^D, we capitalize on the notion of paired-voters presented in the previous section. This allows us to express two new PAC-Bayesian bounds on the risk of a majority vote, one for the supervised case and another for the semi-supervised case.

5.4.1 A PAC-Bayesian Theorem for e_Q^D, s_Q^D, or d_Q^D

The following PAC-Bayesian theorem can either bound the expected disagreement d_Q^D, the expected joint success s_Q^D or the expected joint error e_Q^D of a majority vote (see Definitions 7 and 23).

Theorem 25 *For any distribution D on $\mathcal{X}\times\{-1,1\}$, for any set $\mathcal{H}$ of voters $\mathcal{X}\to[-1,1]$, for any prior distribution P on $\mathcal{H}$, and any $\delta\in(0,1]$, we have*

$$\Pr_{S\sim D^m}\left(\begin{array}{l}\textit{For all posteriors } Q \textit{ on } \mathcal{H}:\\ \mathrm{kl}\big(\alpha_Q^S \,\big\|\, \alpha_Q^D\big) \;\le\; \dfrac{1}{m}\left[2{\cdot}\mathrm{KL}(Q\|P) + \ln\dfrac{\xi(m)}{\delta}\right]\end{array}\right) \;\ge\; 1-\delta\,,$$

where $\alpha_Q^{D'}$ can be either $e_Q^{D'}$, $s_Q^{D'}$ or $d_Q^{D'}$.

Proof Theorem 25 is deduced from Theorem 20. We present here the proof for $\alpha_Q^{D'} = e_Q^{D'}$. The two other cases are very similar.

Consider the set of paired-voters $\mathcal{H}^2$ and the posterior distribution Q^2 of Equation (26).

Also consider the prior distribution P^2 on $\mathcal{H}^2$ such that $P^2(f_{ij}) \stackrel{\text{def}}{=} P(f_i) \cdot P(f_j)$. Then we have,

$$\begin{aligned}
\mathrm{KL}(Q^2 \| P^2) &= \mathop{\mathbf{E}}_{f_{ij}\sim Q^2} \ln \frac{Q^2(f_{ij})}{P^2(f_{ij})} = \mathop{\mathbf{E}}_{f_{ij}\sim Q^2} \ln \frac{Q(f_i)\cdot Q(f_j)}{P(f_i)\cdot P(f_j)} \\
&= \mathop{\mathbf{E}}_{f_{ij}\sim Q^2} \left[\ln \frac{Q(f_i)}{P(f_i)} + \ln \frac{Q(f_j)}{P(f_j)} \right] \\
&= 2 \cdot \mathrm{KL}(Q\|P)\,.
\end{aligned}$$

Finally, from Equation (28), we have $\mathop{\mathbf{E}}_{f_{ij}\sim Q^2} \mathbb{E}_D^{\mathcal{L}_e}\Big(f_{ij}\Big) = e_Q^D$ and $\mathop{\mathbf{E}}_{f_{ij}\sim Q^2} \mathbb{E}_S^{\mathcal{L}_e}\Big(f_{ij}\Big) = e_Q^S$. Hence, by applying Theorem 20, we are done. ■

5.4.2 A New Bound for the Risk of the Majority Vote

Based on the fact that Theorem 25 gives a lower bound on the expected disagreement d_Q^D, we now derive PAC-Bound 1, which is a PAC-Bayesian bound for the $\mathcal{C}$-bound, and therefore, for the risk of the majority vote.

Given any prior distribution P on $\mathcal{H}$, we need the interval $\mathcal{R}_{Q,S}^{\delta}$ of Equation (23), together with

$$\mathcal{D}_{Q,S}^{\delta} \stackrel{\text{def}}{=} \left\{ d \,:\, \mathrm{kl}(d_Q^S \| \, d) \,\leq\, \frac{1}{m}\left[2{\cdot}\mathrm{KL}(Q\|P) + \ln \frac{\xi(m)}{\delta} \right] \right\}. \tag{29}$$

We then express the following bound on the Bayes risk.

PAC-Bound 1 *For any distribution D on $\mathcal{X}\times\{-1,1\}$, for any set $\mathcal{H}$ of voters $\mathcal{X}\to[-1,1]$, for any prior distribution P on $\mathcal{H}$, and any $\delta\in(0,1]$, we have*

$$\Pr_{S\sim D^m}\left(\forall\, Q \text{ on } \mathcal{H} \,:\, R_D(B_Q) \,\leq\, 1 - \frac{\Big(1 - 2\cdot \sup \mathcal{R}_{Q,S}^{\delta/2}\Big)^2}{1 - 2\cdot \inf \mathcal{D}_{Q,S}^{\delta/2}} \right) \geq\, 1-\delta\,,$$

where $\mathcal{R}_{Q,S}^{\delta/2}$ and $\mathcal{D}_{Q,S}^{\delta/2}$ are respectively defined by Equations (23) *and* (29).

Proof By Proposition 9, we have that $d_Q^S \leq \frac{1}{2}$. This, together with the facts that m is finite and $d_Q^S \in \mathcal{D}_{Q,S}^{\delta}$, implies that $\inf \mathcal{D}_{Q,S}^{\delta/2} < \frac{1}{2}$, and therefore that the denominator of the fraction in the statement of PAC-Bound 1 is always strictly positive.

Necessarily, $\sup \mathcal{R}_{Q,S}^{\delta/2} \leq \frac{1}{2}$. Let us consider the two following cases.

Case 1: $\sup \mathcal{R}_{Q,S}^{\delta/2} = \frac{1}{2}$. Then, $1 - 2\cdot\sup\mathcal{R}_{Q,S}^{\delta/2} = 0$, and the bound on $R_D(B_Q)$ is 1, which is trivially valid.

Case 2: $\sup \mathcal{R}_{Q,S}^{\delta/2} < \frac{1}{2}$. Then, we can apply the third form of Theorem 11 to obtain the upper bound on $R_D(B_Q)$. The desired bound is obtained by replacing d_Q^D by its lower bound

$\inf \mathcal{D}_{Q,S}^{\delta/2}$, and $R_D(G_Q)$, by its upper bound $\sup \mathcal{R}_{Q,S}^{\delta/2}$. The two bounds can therefore be deduced by suitably applying Corollary 21 (replacing δ by $\delta/2$) and Theorem 25 (replacing α_Q^S by d_Q^S, α_Q^D by d_Q^D and δ by $\delta/2$). ■

This bound has a major inconvenience: it degrades rapidly if the bounds on the numerator and the denominator are not tight. Note however that in the semi-supervised framework, we can achieve tighter results because the labels of the examples do not affect the value of $d_Q^{D'}$ (see Definition 7). Indeed, it is generally assumed in this framework that the learner has access to a huge amount m' of unlabeled data (*i.e.*, $m' \gg m$). One can then obtain a tighter bound of the disagreement. In this context, PAC-Bound 1' stated below is tighter than PAC-Bound 1.

PAC-Bound 1' (Semi-supervised bound) *For any distribution D on $\mathcal{X}\times\{-1,1\}$, for any set $\mathcal{H}$ of voters $\mathcal{X}\to[-1,1]$, for any prior distribution P on $\mathcal{H}$, and any $\delta\in(0,1]$, we have*

$$\Pr_{\substack{S\sim D^m\\ S_{\mathcal{U}}\sim D_{unlabeled}^{m'}}}\left(\forall Q \text{ on } \mathcal{H} : R_D(B_Q) \leq 1-\frac{\left(1-2\cdot\sup\mathcal{R}_{Q,S}^{\delta/2}\right)^2}{1-2\cdot\inf\mathcal{D}_{Q,S_{\mathcal{U}}}^{\delta/2}}\right)\geq 1-\delta\,.$$

Proof In the presence of a large amount of unlabeled data (denoted by the set $S_{\mathcal{U}}$), one can use Corollary 25 to obtain an accurate lower bound of d_Q^D. An upper bound of $R_D(G_Q)$ can also be obtained via Corollary 21 but, this time, on the labeled data S. Thus, similarly as in the proof of PAC-Bound 1, the result follows from Theorem 11. ■

5.4.3 Computation of PAC-Bounds 1 and 1'

To compute PAC-Bound 1, we obtain the values of $r=\sup\mathcal{R}_{Q,S}^{\delta/2}$ and $d=\inf\mathcal{D}_{Q,S}^{\delta/2}$ by solving

$$\begin{aligned}\mathrm{kl}\big(R_S(G_Q)\,\big\|\,r\big) &= \tfrac{1}{m}\big[\mathrm{KL}(Q\|P)+\ln\tfrac{\xi(m)}{\delta/2}\big]\,, \qquad \text{with } R_S(G_Q)\leq r\leq\tfrac{1}{2}\,,\\ \text{and }\ \mathrm{kl}\big(d_Q^S\,\big\|\,d\big) &= \tfrac{1}{m}\big[2\cdot\mathrm{KL}(Q\|P)+\ln\tfrac{\xi(m)}{\delta/2}\big]\,, \quad \text{with } d\leq d_Q^S\,.\end{aligned}$$

These equations are very similar to the one we solved to compute PAC-Bound 0, as described in Section 5.2.2. Once r and d are computed, the bound on $R_D(B_Q)$ is given by $1-\frac{(1-2\cdot r)^2}{1-2\cdot d}$.

The same methodology can be used to compute PAC-Bound 1', except that in the semi-supervised setting, the disagreement is computed on the unlabeled data $S_{\mathcal{U}}$.

5.5 PAC-Bayesian Theory to Directly Bound the $\mathcal{C}$-bound

PAC-Bounds 1 and 1' of the last section require two approximations to upper bound $\mathcal{C}_Q^D$: one on $R_D(G_Q)$ and another on d_Q^D. We introduce below an extension to the PAC-Bayesian theory (Theorem 28) that enables us to directly bound $\mathcal{C}_Q^D$. To do so, we directly bound any pair of expectations among e_Q^D, s_Q^D and d_Q^D. For this reason, the new PAC-Bayesian theorem

is based on a trivalent random variable instead of a Bernoulli one (which is bivalent). Note that Seeger (2003) and Seldin and Tishby (2010) have presented more general PAC-Bayesian theorems valid for k-valent random variables, for any positive integer k. However, our result leads to tighter bounds for the $k = 3$ case.

Before we get to this new PAC-Bayesian theorem (Theorem 28), we need some preliminary results.

5.5.1 A General PAC-Bayesian Theorem for Two Losses of Paired-Voters

Theorem 26 below allows us to simultaneously bound two losses of paired-voters. This result is inspired by the general PAC-Bayesian theorem for real-valued losses (Theorem 18).

Theorem 26 *For any distribution D on $\mathcal{X}\times\{-1,1\}$, for any set $\mathcal{H}$ of voters $\mathcal{X}\to[-1,1]$, for any two losses $\mathcal{L}_\alpha, \mathcal{L}_\beta : [-1,1]\times\{-1,1\}\to[0,1]$ with $\alpha,\beta\in\{e,s,d\}$, for any prior distribution P on $\mathcal{H}$, for any $\delta\in(0,1]$, for any $m'>0$, and for any convex function $\mathcal{D}(q_1,q_2\,\|\,p_1,p_2)$, we have*

$$\Pr_{S\sim D^m}\left(\begin{array}{l}\textit{For all posteriors } Q \textit{ on } \mathcal{H}:\\ \mathcal{D}\Big(\mathop{\mathbf{E}}_{f_{ij}\sim Q^2}\mathbb{E}_S^{\mathcal{L}_\alpha}\big(f_{ij}\big)\,,\mathop{\mathbf{E}}_{f_{ij}\sim Q^2}\mathbb{E}_S^{\mathcal{L}_\beta}\big(f_{ij}\big)\;\Big\|\;\mathop{\mathbf{E}}_{f_{ij}\sim Q^2}\mathbb{E}_D^{\mathcal{L}_\alpha}\big(f_{ij}\big)\,,\mathop{\mathbf{E}}_{f_{ij}\sim Q^2}\mathbb{E}_D^{\mathcal{L}_\beta}\big(f_{ij}\big)\Big)\\ \qquad\qquad\qquad\qquad\leq\quad \dfrac{1}{m'}\left[2\cdot\mathrm{KL}(Q\|P)+\ln\left(\dfrac{\Omega}{\delta}\right)\right]\end{array}\right)\geq 1-\delta\,,$$

where $\Omega \stackrel{\text{def}}{=} \mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{f_{ij}\sim P^2} e^{m'\cdot\mathcal{D}\left(\mathbb{E}_S^{\mathcal{L}_\alpha}\left(f_{ij}\right),\,\mathbb{E}_S^{\mathcal{L}_\beta}\left(f_{ij}\right)\,\left\|\,\mathbb{E}_D^{\mathcal{L}_\alpha}\left(f_{ij}\right),\,\mathbb{E}_D^{\mathcal{L}_\beta}\left(f_{ij}\right)\right.\right)}$.

Proof To simplify the notation, first let $\alpha_{ij}^{D'}\stackrel{\text{def}}{=}\mathbb{E}_{D'}^{\mathcal{L}_\alpha}\big(f_{ij}\big)$ and $\beta_{ij}^{D'}\stackrel{\text{def}}{=}\mathbb{E}_{D'}^{\mathcal{L}_\beta}\big(f_{ij}\big)$.

Now, since $\mathbf{E}_{f_{ij}\sim P^2} e^{m'\cdot\mathcal{D}\left(\alpha_{ij}^S,\beta_{ij}^S\,\|\,\alpha_{ij}^D,\beta_{ij}^D\right)}$ is a positive random variable, Markov's inequality (Lemma 46, in Appendix A) can be applied to give

$$\Pr_{S\sim D^m}\left(\mathop{\mathbf{E}}_{f_{ij}\sim P^2} e^{m'\cdot\mathcal{D}\left(\alpha_{ij}^S,\beta_{ij}^S\,\|\,\alpha_{ij}^D,\beta_{ij}^D\right)}\leq\frac{1}{\delta}\mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{f_{ij}\sim P^2} e^{m'\cdot\mathcal{D}\left(\alpha_{ij}^S,\beta_{ij}^S\,\|\,\alpha_{ij}^D,\beta_{ij}^D\right)}\right)\geq 1-\delta\,.$$

By exploiting the fact that $\ln(\cdot)$ is an increasing function, and by the definition of Ω, we obtain

$$\Pr_{S\sim D^m}\left(\ln\left[\mathop{\mathbf{E}}_{f_{ij}\sim P^2} e^{m'\cdot\mathcal{D}\left(\alpha_{ij}^S,\beta_{ij}^S\,\|\,\alpha_{ij}^D,\beta_{ij}^D\right)}\right]\;\leq\;\ln\left[\frac{\Omega}{\delta}\right]\right)\geq 1-\delta\,. \tag{30}$$

We apply the change of measure inequality (Lemma 17) on the left side of innermost inequality, with $\phi(f)=m'\cdot\mathcal{D}\big(\alpha_{ij}^S,\beta_{ij}^S\,\|\,\alpha_{ij}^D,\beta_{ij}^D\big)$, $P=P^2$ and $Q=Q^2$. We then use Jensen's

inequality (Lemma 47, in Appendix A), exploiting the convexity of $\mathcal{D}$:

$$\begin{aligned}
&\ln\left[\mathop{\mathbf{E}}_{f_{ij}\sim P^2} e^{m'\cdot\mathcal{D}\left(\alpha^S_{ij},\beta^S_{ij}\,\middle\|\,\alpha^D_{ij},\beta^D_{ij}\right)}\right] \\
&\quad\geq\ m'\mathop{\mathbf{E}}_{f_{ij}\sim Q^2}\mathcal{D}\left(\alpha^S_{ij},\beta^S_{ij}\,\middle\|\,\alpha^D_{ij},\beta^D_{ij}\right)-\mathrm{KL}\left(Q^2\middle\|P^2\right) \\
&\quad\geq\ m'\cdot\mathcal{D}\left(\mathop{\mathbf{E}}_{f_{ij}\sim Q^2}\alpha^S_{ij},\mathop{\mathbf{E}}_{f_{ij}\sim Q^2}\beta^S_{ij}\,\middle\|\,\mathop{\mathbf{E}}_{f_{ij}\sim Q^2}\alpha^D_{ij},\mathop{\mathbf{E}}_{f_{ij}\sim Q^2}\beta^D_{ij}\right)-\mathrm{KL}\left(Q^2\middle\|P^2\right) \\
&\quad=\ m'\cdot\mathcal{D}\left(\mathop{\mathbf{E}}_{f_{ij}\sim Q^2}\alpha^S_{ij},\mathop{\mathbf{E}}_{f_{ij}\sim Q^2}\beta^S_{ij}\,\middle\|\,\mathop{\mathbf{E}}_{f_{ij}\sim Q^2}\alpha^D_{ij},\mathop{\mathbf{E}}_{f_{ij}\sim Q^2}\beta^D_{ij}\right)-2\cdot\mathrm{KL}\left(Q\middle\|P\right).
\end{aligned}$$

The last equality $\mathrm{KL}\left(Q^2\middle\|P^2\right)=2\cdot\mathrm{KL}\left(Q\middle\|P\right)$ has been shown in the proof of Theorem 25. The result can then be straightforwardly obtained by inserting the last inequality into Equation (30). ■

5.5.2 A PAC-Bayesian Theorem for Any Pair Among e^D_Q, s^D_Q, and d^D_Q

In Section 5.1, Theorem 20 was obtained from Theorem 18. Similarly, the main theorem of this subsection (Theorem 28) is deduced from Theorem 26. However, a notable difference between Theorems 20 and 28 is that the former uses of the KL-divergence $\mathrm{kl}(\cdot\|\cdot)$ between distributions of two Bernoulli (*i.e.*, *bivalent*) random variables, and the latter uses the KL-divergence $\mathrm{kl}(\cdot,\cdot\|\cdot,\cdot)$ between distributions of two *trivalent* random variables.

Given two trivalent random variables Y_q and Y_p with $P(Y_q=a)=q_1$, $P(Y_q=b)=q_2$, $P(Y_q=c)=1-q_1-q_2$, and $P(Y_p=a)=p_1$, $P(Y_p=b)=p_2$, $P(Y_p=c)=1-p_1-p_2$, we denote by $\mathrm{kl}(q_1,q_2\,\|\,p_1,p_2)$ the Kullback-Leibler divergence between Y_q and Y_p. Thus, we have

$$\mathrm{kl}(q_1,q_2\,\|\,p_1,p_2)\ \stackrel{\text{def}}{=}\ q_1\ln\frac{q_1}{p_1}+q_2\ln\frac{q_2}{p_2}+(1-q_1-q_2)\ln\frac{1-q_1-q_2}{1-p_1-p_2}\,. \tag{31}$$

Note that $\mathrm{kl}\left(q_1,q_2\,\|\,p_1,p_2\right)$ is a shorthand notation for $\mathrm{KL}(Q\|P)$ of Equation (20), with $Q=(q_1,q_2,1-q_1-q_2)$ and $P=(p_1,p_2,1-p_1-p_2)$. Corollary 50 (in Appendix A) shows that $\mathrm{kl}\left(q_1,q_2\,\|\,p_1,p_2\right)$ is a convex function.

To be able to apply Theorem 26 with $\mathcal{D}(q_1,q_2\,\|\,p_1,p_2)=\mathrm{kl}(q_1,q_2\|p_1,p_2)$, we need Lemma 27 (below). This lemma is inspired by Lemma 19. However, in contrast with the latter, which is based on Maurer's lemma, Lemma 27 needs a generalization of it to trivalent random variables (instead of bivalent ones). The proof of this generalization is provided in Appendix A, listed as Lemma 52.

Lemma 27 *For any distribution D on $\mathcal{X}\times\{-1,1\}$, for any paired-voters f_{ij}, and any positive integer m, we have*

$$\mathop{\mathbf{E}}_{S\sim D^m} e^{m\cdot\mathrm{kl}\left(\mathbb{E}^{\mathcal{L}_\alpha}_S(f_{ij}),\mathbb{E}^{\mathcal{L}_\beta}_S(f_{ij})\ \middle\|\ \mathbb{E}^{\mathcal{L}_\alpha}_D(f_{ij}),\mathbb{E}^{\mathcal{L}_\beta}_D\left(f_{ij}\right)\right)}\ \leq\ \xi(m)+m\,,$$

where $\mathcal{L}_\alpha$ and $\mathcal{L}_\beta$ can be any two of the three losses $\mathcal{L}_s$, $\mathcal{L}_e$ or $\mathcal{L}_d$, and where $\xi(m)$ is defined at Equation (22). Therefore, $m+\sqrt{m}\ \leq\ \xi(m)+m\ \leq\ m+2\sqrt{m}$.

Proof Let Y_{ij} be a random variable that follows a multinomial distribution with three possible outcomes: $a \stackrel{\text{def}}{=} (1,0)$, $b \stackrel{\text{def}}{=} (0,1)$ and $c \stackrel{\text{def}}{=} (0,0)$. The "*Trinomial*" distribution is chosen such that $\Pr\left(Y_{ij}=a\right) = \mathbb{E}_D^{\mathcal{L}_\alpha}(f_{ij})$, $\Pr\left(Y_{ij}=b\right) = \mathbb{E}_D^{\mathcal{L}_\beta}(f_{ij})$ and $\Pr\left(Y_{ij}=c\right) = 1 - \mathbb{E}_D^{\mathcal{L}_\alpha}(f_{ij}) - \mathbb{E}_D^{\mathcal{L}_\beta}(f_{ij})$. Given m trials of Y_{ij}, we denote Y_{ij}^a, Y_{ij}^b and Y_{ij}^c the number of times each outcome is observed. Note that Y_{ij} is totally defined by (Y_{ij}^a, Y_{ij}^b), since $Y_{ij}^c = m - Y_{ij}^a - Y_{ij}^b$. We thus use the notation

$$Y_{ij} \;=\; (Y_{ij}^a, Y_{ij}^b) \;\sim\; \mathcal{T}_{ij} \;\stackrel{\text{def}}{=}\; \text{Trinomial}\Big(m,\, \mathbb{E}_D^{\mathcal{L}_\alpha}(f_{ij}),\, \mathbb{E}_D^{\mathcal{L}_\beta}(f_{ij})\Big).$$

Hence, we have

$$\Pr_{(Y_{ij}^a, Y_{ij}^b)\sim\mathcal{T}_{ij}}\Big(Y_{ij}^a{=}k_1 \wedge Y_{ij}^b{=}k_2\Big) = \tbinom{m}{k_1}\tbinom{m-k_1}{k_2}\big[\mathbb{E}_S^{\mathcal{L}_\alpha}(f_{ij})\big]^{k_1}\big[\mathbb{E}_S^{\mathcal{L}_\beta}(f_{ij})\big]^{k_2}\big[1{-}\mathbb{E}_S^{\mathcal{L}_\alpha}(f_{ij}){-}\mathbb{E}_S^{\mathcal{L}_\beta}(f_{ij})\big]^{m-k_1-k_2},$$

for any $k_1 \in \{0,..,m\}$ and any $k_2 \in \{0,..,m{-}k_1\}$.

Now, applying Lemma 52 to the convex function $e^{m\cdot \mathrm{kl}\left(\cdot,\cdot \,\|\, \mathbb{E}_D^{\mathcal{L}_\alpha}(f_{ij}),\mathbb{E}_D^{\mathcal{L}_\beta}(f_{ij})\right)}$, and by the definition of $\mathrm{kl}(\cdot,\cdot\|\cdot,\cdot)$, we have

$$\begin{aligned}
&\mathop{\mathbf{E}}_{S\sim D^m} e^{m\cdot \mathrm{kl}\left(\mathbb{E}_S^{\mathcal{L}_\alpha}(f_{ij}),\mathbb{E}_S^{\mathcal{L}_\beta}(f_{ij}) \,\middle\|\, \mathbb{E}_D^{\mathcal{L}_\alpha}(f_{ij}),\mathbb{E}_D^{\mathcal{L}_\beta}\left(f_{ij}\right)\right)} \\
&\quad\le \mathop{\mathbf{E}}_{(Y_{ij}^a,Y_{ij}^b)\sim\mathcal{T}_{ij}} e^{m\cdot \mathrm{kl}\left(\frac{1}{m}Y_{ij}^a,\frac{1}{m}Y_{ij}^b \,\middle\|\, \mathbb{E}_D^{\mathcal{L}_\alpha}(f_{ij}),\mathbb{E}_D^{\mathcal{L}_\beta}\left(f_{ij}\right)\right)} \\
&\quad= \mathop{\mathbf{E}}_{(Y_{ij}^a,Y_{ij}^b)\sim\mathcal{T}_{ij}} \left(\frac{\frac{1}{m}Y_{ij}^a}{\mathbb{E}_S^{\mathcal{L}_\alpha}(f_{ij})}\right)^{Y_{ij}^a}\left(\frac{\frac{1}{m}Y_{ij}^b}{\mathbb{E}_S^{\mathcal{L}_\beta}(f_{ij})}\right)^{Y_{ij}^b}\left(\frac{1-\frac{1}{m}Y_{ij}^a-\frac{1}{m}Y_{ij}^b}{1-\mathbb{E}_S^{\mathcal{L}_\alpha}(f_{ij})-\mathbb{E}_S^{\mathcal{L}_\beta}(f_{ij})}\right)^{m-Y_{ij}^a-Y_{ij}^b}.
\end{aligned}$$

As Y_{ij} follows a trinomial law, we then have

$$\begin{aligned}
&\mathop{\mathbf{E}}_{(Y_{ij}^a,Y_{ij}^b)\sim\mathcal{T}_{ij}} \left(\frac{\frac{1}{m}Y_{ij}^a}{\mathbb{E}_S^{\mathcal{L}_\alpha}(f_{ij})}\right)^{Y_{ij}^a}\left(\frac{\frac{1}{m}Y_{ij}^b}{\mathbb{E}_S^{\mathcal{L}_\beta}(f_{ij})}\right)^{Y_{ij}^b}\left(\frac{1-\frac{1}{m}Y_{ij}^a-\frac{1}{m}Y_{ij}^b}{1-\mathbb{E}_S^{\mathcal{L}_\alpha}(f_{ij})-\mathbb{E}_S^{\mathcal{L}_\beta}(f_{ij})}\right)^{m-Y_{ij}^a-Y_{ij}^b} \\
&\quad= \sum_{k_1=0}^{m}\sum_{k_2=0}^{m-k_1}\left[\Pr_{(Y_{ij}^a,Y_{ij}^b)\sim\mathcal{T}_{ij}}\Big(Y_{ij}^a = k_1 \wedge Y_{ij}^b = k_2\Big)\right. \\
&\qquad\qquad\left.\times\left(\frac{\frac{k_1}{m}}{\mathbb{E}_S^{\mathcal{L}_\alpha}(f_{ij})}\right)^{k_1}\left(\frac{\frac{k_2}{m}}{\mathbb{E}_S^{\mathcal{L}_\beta}(f_{ij})}\right)^{k_2}\left(\frac{1-\frac{k_1}{m}-\frac{k_2}{m}}{1-\mathbb{E}_S^{\mathcal{L}_\alpha}(f_{ij})-\mathbb{E}_S^{\mathcal{L}_\beta}(f_{ij})}\right)^{m-k_1-k_2}\right] \\
&\quad= \sum_{k_1=0}^{m}\sum_{k_2=0}^{m-k_1}\left[\binom{m}{k_1}\binom{m-k_1}{k_2}\Big(\mathbb{E}_S^{\mathcal{L}_\alpha}(f_{ij})\Big)^{k_1}\Big(\mathbb{E}_S^{\mathcal{L}_\beta}(f_{ij})\Big)^{k_2}\Big(1{-}\mathbb{E}_S^{\mathcal{L}_\alpha}(f_{ij}){-}\mathbb{E}_S^{\mathcal{L}_\beta}(f_{ij})\Big)^{m-k_1-k_2}\right. \\
&\qquad\qquad\left.\times\left(\frac{\frac{k_1}{m}}{\mathbb{E}_S^{\mathcal{L}_\alpha}(f_{ij})}\right)^{k_1}\left(\frac{\frac{k_2}{m}}{\mathbb{E}_S^{\mathcal{L}_\beta}(f_{ij})}\right)^{k_2}\left(\frac{1-\frac{k_1}{m}-\frac{k_2}{m}}{1-\mathbb{E}_S^{\mathcal{L}_\alpha}(f_{ij})-\mathbb{E}_S^{\mathcal{L}_\beta}(f_{ij})}\right)^{m-k_1-k_2}\right] \\
&\quad= \sum_{k_1=0}^{m}\sum_{k_2=0}^{m-k_1}\binom{m}{k_1}\binom{m-k_1}{k_2}\left(\frac{k_1}{m}\right)^{k_1}\left(\frac{k_2}{m}\right)^{k_2}\left(1-\frac{k_1}{m}-\frac{k_2}{m}\right)^{m-k_1-k_2} \\
&\quad= \xi(m) + m\,.
\end{aligned}$$

The last equality has been proven by Younsi (2012). Recall that $\xi(m)$ is defined by Equation (22). ■

We are now ready to present the main result of this section. By bounding any pair of expectations among e_Q^D, s_Q^D and d_Q^D, Theorem 28 is the perfect tool to directly bound the $\mathcal{C}$-bound.

Theorem 28 *For any distribution D on $\mathcal{X}\times\{-1,1\}$, for any set $\mathcal{H}$ of voters $\mathcal{X}\to[-1,1]$, for any prior distribution P on $\mathcal{H}$, and any $\delta\in(0,1]$, we have*

$$\Pr_{S\sim D^m}\left(\begin{array}{l}\textit{For all posteriors } Q \textit{ on } \mathcal{H}:\\ \mathrm{kl}\big(\,\alpha_Q^S,\beta_Q^S\,\big\|\,\alpha_Q^D,\beta_Q^D\,\big)\;\le\;\dfrac{1}{m}\left[2{\cdot}\mathrm{KL}(Q\|P)+\ln\dfrac{\xi(m)+m}{\delta}\right]\end{array}\right)\ge 1-\delta\,,$$

where $\alpha_Q^{D'}$ and $\beta_Q^{D'}$ can be any two distinct choices among $d_Q^{D'}$, $e_Q^{D'}$ and $s_Q^{D'}$.

Proof The result follows from Theorem 26 with $\mathcal{D}(q_1,q_2\,\|\,p_1,p_2)=\mathrm{kl}(q_1,q_2\,\|\,p_1,p_2)$ and $m'=m$. Since Equation (28) shows that $\alpha_Q^{D'}=\mathop{\mathbf{E}}_{f_{ij}\sim Q^2}\alpha_{ij}^{D'}$ and $\beta_Q^{D'}=\mathop{\mathbf{E}}_{f_{ij}\sim Q^2}\beta_{ij}^{D'}$, we have

$$\Pr_{S\sim D^m}\Bigg(\forall\,Q\text{ on }\mathcal{H}\colon\;\mathrm{kl}\Big(\alpha_Q^S,\beta_Q^S\,\|\,\alpha_Q^D,\beta_Q^D\Big)\le \frac{1}{m}\left[2\cdot\mathrm{KL}(Q\|P)+\ln\left(\frac{1}{\delta}\mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{f_{ij}\sim P^2}e^{m\mathrm{kl}\left(\alpha_{ij}^S,\beta_{ij}^S\,\middle\|\,\alpha_{ij}^D,\beta_{ij}^D\right)}\right)\right]\Bigg)\ge 1-\delta\,.$$

As the prior distribution P^2 is independent of S, we can swap the two expectations in expression $\mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{f_{ij}\sim P^2}e^{m\mathrm{kl}(\alpha_{ij}^S,\beta_{ij}^S\,\|\,\alpha_{ij}^D,\beta_{ij}^D)}$. This observation, together with Lemma 27, gives

$$\begin{aligned}\mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{f_{ij}\sim P^2}e^{m\mathrm{kl}\left(\alpha_{ij}^S,\beta_{ij}^S\,\middle\|\,\alpha_{ij}^D,\beta_{ij}^D\right)} &= \mathop{\mathbf{E}}_{f_{ij}\sim P^2}\mathop{\mathbf{E}}_{S\sim D^m}e^{m\mathrm{kl}\left(\alpha_{ij}^S,\beta_{ij}^S\,\middle\|\,\alpha_{ij}^D,\beta_{ij}^D\right)}\\ &\le \mathop{\mathbf{E}}_{f_{ij}\sim P^2}\xi(m)+m\\ &= \xi(m)+m\,.\end{aligned}$$

■

A first version of Theorem 28 was proposed by Lacasse et al. (2006), with the difference that $\ln\frac{(m+1)(m+2)}{2\delta}$ in the latter is now replaced by $\ln\frac{\xi(m)+m}{\delta}$ in the former. Since $\xi(m)+m<\frac{(m+1)(m+2)}{2}$, the new theorem is therefore tighter.

5.5.3 Another Bound for the Risk of the Majority Vote

First, we need the following notation that is related to Theorem 28. Given any prior distribution P on $\mathcal{H}$,

$$\mathcal{A}_{Q,S}^{\delta}\;\stackrel{\text{def}}{=}\;\left\{(d,e)\;:\;\mathrm{kl}(d_Q^S,e_Q^S\|\,d,e)\;\le\;\frac{1}{m}\left[2{\cdot}\mathrm{KL}(Q\|P)+\ln\tfrac{\xi(m)+m}{\delta}\right]\right\}. \qquad (32)$$

The bound is obtained by seeking the point of $\mathcal{A}_{Q,S}^{\delta}$ maximizing the $\mathcal{C}$-bound. Since a point (d,e) of $\mathcal{A}_{Q,S}^{\delta}$ expresses a disagreement d and a joint error e, we directly compute the bound on $\mathcal{C}_Q^D$ using Equation (25).

Note however that $\mathcal{A}^{\delta}_{Q,S}$ can contain points that are not possible in practice, *i.e.*, points that are not achievable with any data-generating distribution D. Indeed, by Proposition 9, we know that

$$d^D_Q \;\leq\; 2 \cdot R_D(G_Q) \cdot \big(1 - R_D(G_Q)\big)\,.$$

Based on this property, it is possible to significantly reduce the achievable region of $\mathcal{A}^{\delta}_{Q,S}$. To do so, we must first rewrite this property based on d^D_Q and e^D_Q only.

$$\begin{array}{rrcl} & d^D_Q & \leq & 2 \cdot R_D(G_Q) \cdot \big(1 - R_D(G_Q)\big) \;=\; 2 \cdot \big(e^D_Q + \frac{1}{2} d^D_Q\big) \cdot \big(1 - (e^D_Q + \frac{1}{2} d^D_Q)\big) \\ \Leftrightarrow & 0 & \leq & -\frac{1}{2}(d^D_Q)^2 - 2e^D_Q \cdot d^D_Q + 2e^D_Q - 2(e^D_Q)^2 \\ \Leftrightarrow & d^D_Q & \leq & 2 \cdot \Big(\sqrt{e^D_Q} - e^D_Q\Big)\,. \end{array} \tag{33}$$

Note also that if $R_D(G_Q) \geq \frac{1}{2}$, there is no bound on $R_D(B_Q)$ better than the trivial one $R_D(B_Q) \leq 1$. We therefore consider only the pairs $(d, e) \in \mathcal{A}^{\delta}_{Q,S}$ that do not correspond to that situation. Since $R_D(G_Q) = \frac{1}{2}(2e^D_Q + d^D_Q)$ (Equation 24), this is therefore equivalent to considering only the pairs (d, e) such that $2e + d < 1$. We later show that this still gives a valid bound. Thus, from all these ideas, we restrain $\mathcal{A}^{\delta}_{Q,S}$ (Equation 32) as follows:

$$\widetilde{\mathcal{A}}^{\delta}_{Q,S} \;\stackrel{\text{def}}{=}\; \left\{(d, e) \in \mathcal{A}^{\delta}_{Q,S} \;:\; d \,\leq\, 2(\sqrt{e} - e) \quad \text{and} \quad 2e + d < 1\right\}, \tag{34}$$

and obtain the following bound that, in contrast with PAC-Bound 1, directly bounds $\mathcal{C}^D_Q$.

PAC-Bound 2 *For any distribution D on $\mathcal{X} \times \{-1, 1\}$, for any set $\mathcal{H}$ of voters $\mathcal{X} \to [-1, 1]$, for any prior distribution P on $\mathcal{H}$, and any $\delta \in (0, 1]$, we have*

$$\Pr_{S \sim D^m}\left(\forall\, Q \text{ on } \mathcal{H} \;:\; R_D(B_Q) \,\leq \sup_{(d,e) \in \widetilde{\mathcal{A}}^{\delta}_{Q,S}} \left[1 - \frac{\Big(1 - (2e + d)\Big)^2}{1 - 2d}\right]\right) \geq\, 1 - \delta\,.$$

Proof We need to show that the supremum value in the statement of PAC-Bound 2 is a valid upper bound of $R_D(B_Q)$. Note that if $\widetilde{\mathcal{A}}^{\delta}_{Q,S} = \emptyset$, then the supremum is $+\infty$, and the bound is trivially valid. Therefore, we assume below that $\widetilde{\mathcal{A}}^{\delta}_{Q,S}$ is not empty.

Let us consider $(d, e) \in \widetilde{\mathcal{A}}^{\delta}_{Q,S}$. From the conditions $d \,\leq\, 2(\sqrt{e} - e)$ and $2e + d < 1$, it follows by straightforward calculations that $d < \frac{1}{2}$. This implies that

$$1 - \frac{\Big(1 - (2e + d)\Big)^2}{1 - 2d} \;<\; 1\,,$$

because both the numerator and the denominator of the fraction are strictly positive (remember that $2e + d < 1$). Thus, the supremum is at most 1.

Let us consider the three following cases.

Case 1: The supremum is not attained in $\widetilde{\mathcal{A}}^{\delta}_{Q,S}$. Note that as $\widetilde{\mathcal{A}}^{\delta}_{Q,S}$ is a subset of $\mathbb{R}^2$, the supremum must be attained for a pair in the closure of $\widetilde{\mathcal{A}}^{\delta}_{Q,S}$. The latter is not a closed set only because of its $2e + d < 1$ constraint. Therefore, the supremum is achieved for a pair (d, e) in the closure for which $1 - (2e + d) = 0$, implying that the value of the supremum is in that case 1, which trivially is a valid bound for $R_D(B_Q)$.

Case 2: The supremum is attained in $\widetilde{\mathcal{A}}^{\delta}_{Q,S}$ *and has value* 1. In that case, the bound is again trivially valid.

Case 3: The supremum is attained in $\widetilde{\mathcal{A}}^{\delta}_{Q,S}$ *and has a value strictly lower than* 1. In that case, there must be an $\epsilon > 0$ such that $2e + d < 1 - \epsilon$ for all $(d, e) \in \widetilde{\mathcal{A}}^{\delta}_{Q,S}$. Hence, because of Equation (33) and Theorem 28, we have that $2e^D_Q + d^D_Q < 1 - \epsilon$ with probability $1-\delta$. Since $R_D(G_Q) = \frac{1}{2}(2e^D_Q + d^D_Q)$ (Equation 24), this implies that, with probability $1-\delta$, $R_D(G_Q) < 1/2 - 1/2\epsilon$. Hence, with probability $1-\delta$, Theorem 11 is valid – *i.e.*, $\mathcal{C}^D_Q$ bounds $R_D(B_Q)$ – and $(d^D_Q, e^D_Q) \in \widetilde{\mathcal{A}}^{\delta}_{Q,S}$. Thus,

$$R_D(B_Q) \;\leq\; \mathcal{C}^D_Q \;=\; 1 - \frac{\Big(1 - (2e^D_Q + d^D_Q)\Big)^2}{1 - 2d^D_Q} \;\leq\; \sup_{(d,e)\in\widetilde{\mathcal{A}}^{\delta}_{Q,S}} \left[1 - \frac{\Big(1 - (2e + d)\Big)^2}{1 - 2d} \right],$$

and we are done. ■

In some situations, we can slightly improve PAC-Bound 2 by bounding the joint error e^D_Q via Theorem 25 with δ replaced by $\delta/2$. This removes all pairs (d, e) such that e does not belong to the set $\mathcal{E}^{\delta/2}_{Q,S}$ defined as

$$\mathcal{E}^{\delta/2}_{Q,S} \;\stackrel{\text{def}}{=}\; \left\{ e \;:\; \mathrm{kl}(e^S_Q \| \, e) \;\leq\; \frac{1}{m}\left[2{\cdot}\mathrm{KL}(Q\|P) + \ln \tfrac{\xi(m)}{\delta/2} \right] \right\}.$$

Then, by applying PAC-Bound 2, with δ replaced by $\delta/2$, one can obtain the following slightly improved bound.

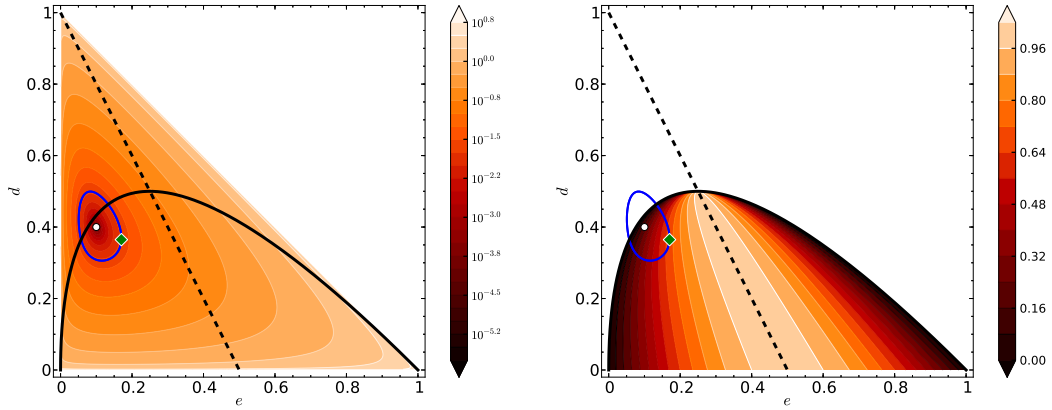
PAC-Bound 2' *For any distribution* D *on* $\mathcal{X}\times\{-1,1\}$*, for any set* $\mathcal{H}$ *of voters* $\mathcal{X} \to [-1,1]$*, for any prior distribution* P *on* $\mathcal{H}$*, and any* $\delta \in (0,1]$*, we have*

$$\Pr_{S\sim D^m}\left(\forall\, Q \text{ on } \mathcal{H} \;:\; R_D(B_Q) \;\leq\; \sup_{(d,e)\in\widehat{\mathcal{A}}^{\delta/2}_{Q,S}} \left[1 - \frac{\Big(1 - (2e + d)\Big)^2}{1 - 2d} \right] \right) \geq\; 1 - \delta\,,$$

where

$$\widehat{\mathcal{A}}^{\delta/2}_{Q,S} \;\stackrel{\text{def}}{=}\; \left\{ (d,e) \in \mathcal{A}^{\delta/2}_{Q,S} \;:\; d \leq 2(\sqrt{e} - e)\,,\;\; 2e + d < 1 \;\text{ and }\; e \leq \sup \mathcal{E}^{\delta/2}_{Q,S} \right\}. \qquad (35)$$

Proof Immediate consequence of Theorem 25, PAC-Bound 2, and the union bound. ■



(a) Contour plot of $\mathrm{kl}(0.4, 0.1 \| d, e)$.

(b) Contour plot of $F_{\mathcal{C}}(d, e)$.

Figure 5: Example of application of PAC-Bound 2. We suppose that $\mathrm{KL}(Q\|P) = 5$, $m = 1000$ and $\delta = 0.05$. If we observe an empirical joint error $e_Q^S = 0.10$ and an empirical disagreement $d_Q^S = 0.40$ (thus, a Gibbs risk $R_S(G_Q) = 0.1 + \frac{1}{2} \cdot 0.4 = 0.30$), then we need to maximize the function $F_{\mathcal{C}}(d, e)$ over the domain $\widetilde{\mathcal{A}}_{Q,S}^{\delta}$ given by three constraints: $\mathrm{kl}(0.4, 0.1 \| \, d, e) \leq \frac{1}{m}\left[2\mathrm{KL}(Q\|P) + \ln \frac{\xi(m)+m}{\delta}\right] \approx 0.0199$ (blue oval), $d \leq 2(\sqrt{e} - e)$ (black curve) and $2e + d < 1$ (black dashed line). Therefore, we obtain a bound $R_D(B_Q) \leq 0.679$ (corresponding to the green diamond marker).

5.5.4 Computation of PAC-Bounds 2 and 2'

Let us consider the $\mathcal{C}$-bound as a function $F_{\mathcal{C}}$ of two variables $(d, e) \in [0, \frac{1}{2}] \times [0, 1]$, instead of a function of the distribution Q.

$$F_{\mathcal{C}}(d, e) \stackrel{\text{def}}{=} 1 - \frac{\left[1 - (2e + d)\right]^2}{1 - 2d}\,. \tag{36}$$

Proposition 54 (provided in Appendix A) shows that $F_{\mathcal{C}}$ is a concave function. Therefore, PAC-Bound 2 is obtained by maximizing $F_{\mathcal{C}}(d, e)$ in the domain $\widetilde{\mathcal{A}}_{Q,S}^{\delta}$ (Equation 34), which is both bounded and convex. Several optimization methods can achieve this. In our experiments, we decompose $F_{\mathcal{C}}(d, e)$ in two nested functions of a single argument:

$$\sup_{(d,e)\in\widetilde{\mathcal{A}}_{Q,S}^{\delta}} \Big[F_{\mathcal{C}}(d, e)\Big] \;=\; \sup_{d:(d,\cdot)\in\widetilde{\mathcal{A}}_{Q,S}^{\delta}} \Big[F_{\mathcal{C}}^*(d)\Big], \quad \text{where } F_{\mathcal{C}}^*(d) \stackrel{\text{def}}{=} \sup_{e:(d,e)\in\widetilde{\mathcal{A}}_{Q,S}^{\delta}} \Big[F_{\mathcal{C}}(d, e)\Big]\,.$$

Thus, we implement the maximization of $F_{\mathcal{C}}$ using a one-dimensional optimization algorithm twice. Figure 5 shows an application example of PAC-Bound 2.

The computation of PAC-Bound 2' is done using the same method, but we optimize over the domain $\widehat{\mathcal{A}}_{Q,S}^{\delta/2}$ (Equation 35) instead of $\widetilde{\mathcal{A}}_{Q,S}^{\delta}$, which is also bounded and convex. Of course, this requires computing $\sup \mathcal{E}_{Q,S}^{\delta/2}$ beforehand, using the same technique as for PAC-Bounds 0, 1 and 1'. Figure 6 shows an application example of PAC-Bound 2'.

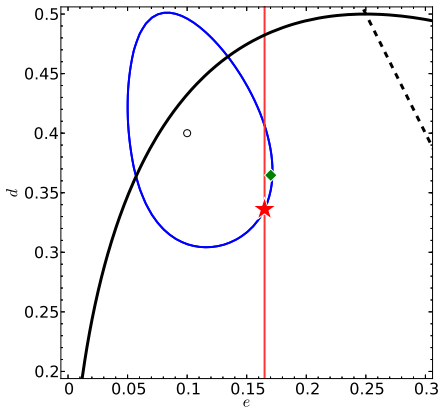


Figure 6: Example of application of PAC-Bound 2'. We use the same quantities as for Figure 5. The red vertical line corresponds to the upper bound on the joint error, resulting in an improved bound of $R_D(B_Q) \leq 0.660$ (corresponding to the red star marker). Note however that, even if the bound here is tighter, the egg-region is a bit bigger than in the case of PAC-Bound 2 because all the δ has been replaced by $\delta/2$.

5.6 Empirical Comparison Between PAC-Bounds on the Bayes Risk $R_D(B_Q)$

We now propose an empirical comparison of all PAC-Bounds we presented so far. The numerical results of Figure 7 are obtained by using AdaBoost (Schapire and Singer, 1999) with decision stumps on the Mushroom UCI data set (which contains 8124 examples). This data set is randomly split into two halves: one training set S and one testing set T. For each round of boosting, we compute the usual PAC-Bayesian bound of twice the Gibbs risk (PAC-Bound 0) of the corresponding majority vote classifier, as well as the other variants of the PAC-Bayesian bounds presented in this paper.

We can see that PAC-Bound 1 is generally tighter than PAC-Bound 0, and we obtain a substantial improvement with PAC-Bound 2. Almost no improvement is obtained with PAC-Bound 2' in that case. We can also see that using unlabeled data to estimate d_Q^D helps, as PAC-Bound 1' is the tightest.[10]

However, we see in Figure 7 that after 8 rounds of boosting, all the bounds are degrading even if the value of $\mathcal{C}_Q^S$ continues to decrease. This drawback is due to the fact that the denominator of $\mathcal{C}_Q^S$ tends to 0, that is the second moment of the margin $\mu_2(M_Q^S)$ is close to 0 (see the first or the second forms of Theorem 11). Hence, in this context, the first moment of the margin $\mu_1(M_Q^S)$ must be small as well. Thus, any slack in the bound of $\mu_1(M_Q^D)$ has a multiplicative effect on each of the three proposed PAC-bounds of $R_D(B_Q)$. Unfortunately, Boosting algorithms tend to construct majority votes with $\mu_1(M_Q^S)$ just slightly larger than 0.

10. To obtain PAC-Bound 1', we simulate the case where we have access to a large number of unlabeled data by simply using the empirical value of d_Q^T computed on the testing set.

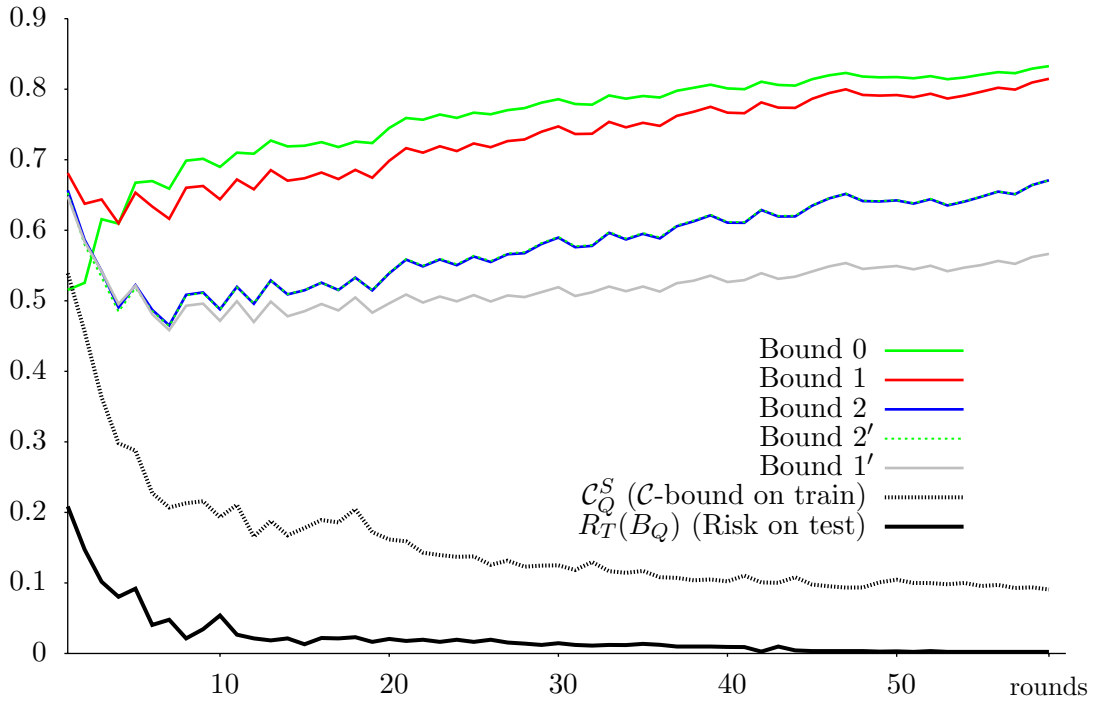


Figure 7: Comparison of bounds of $R_D(B_Q)$ during 60 rounds of Boosting.

6. PAC-Bayesian Bounds without KL

Having PAC-Bayesian theorems that bound the difference between $\mathcal{C}_Q^S$ and $\mathcal{C}_Q^D$ opens the way to structural $\mathcal{C}$-bound minimization algorithms. As for most PAC-Bayesian results, the bound on $\mathcal{C}_Q^D$ depends on an empirical estimate of it, and on the Kullback-Leibler divergence $\mathrm{KL}(Q\|P)$ between the output distribution Q and the a priori defined distribution P. In this section, we present a theoretical extension of our PAC-Bayesian approach that is mandatory to develop the $\mathcal{C}_Q^D$-minimization algorithm of Section 8.

The next theorems introduce PAC-Bayesian bounds that have the surprising property of having no KL term. This new approach is driven by the fact that our attempts to construct algorithms that minimize any of the PAC-Bounds presented in the previous section ended up being unsuccessful. Surprisingly, the KL-divergence is a poor regularizer in this case, as its empirical value tends to be overweighted in comparison with the empirical value of the $\mathcal{C}$-bound (*i.e.*, $\mathcal{C}_Q^S$).

There have already been some attempts to develop PAC-Bayesian bounds that do not rely on the KL-divergence (see the localized priors of Catoni, 2007, or the distribution-dependent priors of Lever et al., 2013). The usual idea is to bound the KL-divergence via some concentration inequality. In the following, the KL term simply vanishes from the bound, provided that we restrict ourselves to *aligned posteriors*, a notion that is properly defined later on in this section. The fact that these new PAC-Bayesian bounds do not

contain any KL divergence terms indicates that the restriction to aligned posteriors has some "built in" regularization action.

The following theory is similar to the one used by Germain et al. (2011), in which two learning algorithms inspired by the PAC-Bayesian theory are compared: one regularized with the KL divergence, using a hyperparameter to control its weight, and one regularized by restricting the posterior distributions to be *aligned* on the prior distribution. Surprisingly, the latter algorithm uses one less parameter, and has been shown to have an as good accuracy.

6.1 Self-Complemented Sets of Voters and Aligned Distributions

In this section, we assume that the (possibly infinite) set of voters $\mathcal{H}$ is *self-complemented*[11].

Definition 29 A set of voters $\mathcal{H}$ is said to be *self-complemented* if there exists a bijection $c : \mathcal{H} \to \mathcal{H}$ such that for any $f \in \mathcal{H}$,

$$c(f) \;=\; -f\,.$$

Moreover, we say that a distribution Q on any self-complemented $\mathcal{H}$ is *aligned* on a prior distribution P if

$$Q(f) + Q(c(f)) \;=\; P(f) + P(c(f))\,, \quad \forall f \in \mathcal{H}\,.$$

When P is the uniform prior distribution and Q is aligned on P, we say that Q is *quasi-uniform*. Note that the uniform distribution is itself a quasi-uniform distribution.

In the finite case, we consider self-complemented sets $\mathcal{H}$ of $2n$ voters $\mathcal{X} \to \overline{\mathcal{Y}}$. In this setting, for any $x \in \mathcal{X}$ and any $i \in \{1, \ldots, n\}$, we have that $f_{i+n}(x) = -f_i(x)$. Moreover, finite quasi-uniform distributions Q is such that for any $i \in \{1, \ldots, n\}$,

$$Q(f_i) + Q(f_{i+n}) \;=\; \frac{1}{n}\,. \tag{37}$$

Equation (37) shows that when a distribution Q is restricted to being quasi-uniform, the sum of the weight given to a pair of complementary voters is equal to $\frac{1}{n}$. As Q is a distribution, this means that the weight of any voter is lower-bounded by 0 and upper-bounded by $\frac{1}{n}$, giving rise to an L_∞-norm regularization. Note that, in this context, the maximum value of $\mathrm{KL}(Q\|P)$ is reached when all voters have a weight of either 0 or $\frac{1}{n}$. Indeed, a quasi-uniform distribution Q is such that $\mathrm{KL}(Q\|P) \;\leq\; n(\frac{1}{n}) \ln(\frac{1}{n}/\frac{1}{2n}) \;=\; \ln 2$. Consequently, the value of the KL term is necessarily small and plays a little role in PAC-Bayesian bounds computed with quasi-uniform distributions. The following theorems and corollaries are specializations that allow to slightly improve these PAC-Bayesian bounds by getting rid of the KL term completely. To achieve these results, the associated proofs require restrictions on the choice of convex function $\mathcal{D}$ and loss function $\mathcal{L}$.

11. In Laviolette et al. (2011), this notion was introduced as an *auto-complemented* set of voters. However, *self-complemented* is a more suitable name. Also, note that a similar notion, called a *symmetric hypothesis class*, is introduced in Daniely et al. (2013).

6.2 PAC-Bayesian Theorems without KL for the Gibbs Risk

Let us first specialize Theorem 18 to aligned distributions and linear loss $\mathcal{L}_\ell$. We first need a new change of measure inequality, as this is the part of Theorem 18 where the KL term appears.

Lemma 30 (Change of measure inequality for aligned posteriors) *For any self-complemented set $\mathcal{H}$, for any distribution P on $\mathcal{H}$, any distribution Q aligned on P, and for any measurable function $\phi : \mathcal{H} \to \mathbb{R}$ such that $\phi(f) = \phi(c(f))$ for all $f \in \mathcal{H}$, we have*

$$\mathop{\mathbf{E}}_{f\sim Q} \phi(f) \;\leq\; \ln\left(\mathop{\mathbf{E}}_{f\sim P} e^{\phi(f)}\right).$$

Proof First, note that one can change the expectation over Q to an expectation over P, using the fact that $\phi(f) = \phi(c(f))$ for any $f \in \mathcal{H}$, and that Q is aligned on P.

$$\begin{aligned}
2 \cdot \mathop{\mathbf{E}}_{f\sim Q} \phi(f) &= \int_{\mathcal{H}} df\; Q(f)\,\phi(f) + \int_{\mathcal{H}} df\; Q(c(f))\,\phi(c(f)) \\
&= \int_{\mathcal{H}} df\; Q(f)\,\phi(f) + \int_{\mathcal{H}} df\; Q(c(f))\,\phi(f) \\
&= \int_{\mathcal{H}} df\; \Big(Q(f) + Q(c(f))\Big)\,\phi(f) \\
&= \int_{\mathcal{H}} df\; \Big(P(f) + P(c(f))\Big)\,\phi(f) \\
&= \int_{\mathcal{H}} df\; P(f)\,\phi(f) + \int_{\mathcal{H}} df\; P(c(f))\,\phi(f) \\
&= \int_{\mathcal{H}} df\; P(f)\,\phi(f) + \int_{\mathcal{H}} df\; P(c(f))\,\phi(c(f)) \\
&= 2 \cdot \mathop{\mathbf{E}}_{f\sim P} \phi(f)\,.
\end{aligned}$$

The result is obtained by changing the expectation over Q to an expectation over P, and then by applying Jensen's inequality (Lemma 47, in Appendix A).

$$\mathop{\mathbf{E}}_{f\sim Q} \phi(f) \;=\; \mathop{\mathbf{E}}_{f\sim P} \phi(f) \;=\; \mathop{\mathbf{E}}_{f\sim P} \ln e^{\phi(f)} \;\leq\; \ln\left(\mathop{\mathbf{E}}_{f\sim P} e^{\phi(f)}\right).$$

■

Theorem 31 (PAC-Bayesian theorem for aligned posteriors) *For any distribution D on $\mathcal{X}\times\{-1,1\}$, any self-complemented set $\mathcal{H}$ of voters $\mathcal{X} \to [-1,1]$, any prior distribution P on $\mathcal{H}$, any convex function $\mathcal{D} : [0,1]\times[0,1] \to \mathbb{R}$ for which $\mathcal{D}(q,p) = \mathcal{D}(1-q,1-p)$, for any $m' > 0$ and any $\delta \in (0,1]$, we have*

$$\Pr_{S\sim D^m}\left(\begin{array}{l}\textit{For all posteriors } Q \textit{ aligned on } P: \\ \mathcal{D}\big(R_S(G_Q), R_D(G_Q)\big) \;\leq\; \dfrac{1}{m'}\left[\ln\left(\dfrac{1}{\delta}\mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{f\sim P} e^{\,m'\cdot\mathcal{D}\left(\mathbb{E}_S^{\mathcal{L}_\ell}(f),\,\mathbb{E}_D^{\mathcal{L}_\ell}(f)\right)}\right)\right]\end{array}\right) \geq\; 1-\delta\,.$$

Similarly to Theorem 18, the statement of Theorem 31 above contains a value m' which is likely to be set to m in most cases. However, the distinction between m and m' is mandatory to develop the PAC-Bayesian theory for sample-compressed voters in Section 7. Indeed, in proofs of forthcoming Theorems 39, 41 and 42, we have $m' = m - \lambda$, where λ is the size of the voters compression sequence (this concept is properly defined in Section 7).

Proof The proof follows the exact same steps as the proof of Theorem 18, using the linear loss $\mathcal{L} = \mathcal{L}_\ell$ and replacing the use of the change of measure inequality (Lemma 17) by the change of measure inequality for aligned posteriors (Lemma 30), with $\phi(f) = m' \cdot \mathcal{D}\left(\mathbb{E}_S^{\mathcal{L}_\ell}(f), \mathbb{E}_D^{\mathcal{L}_\ell}(f)\right)$. Note that this function has the required property, as

$$\mathcal{D}\left(\mathbb{E}_S^{\mathcal{L}_\ell}(f), \mathbb{E}_D^{\mathcal{L}_\ell}(f)\right) = \mathcal{D}\left(1 - \mathbb{E}_S^{\mathcal{L}_\ell}(c(f)), 1 - \mathbb{E}_D^{\mathcal{L}_\ell}(c(f))\right) = \mathcal{D}\left(\mathbb{E}_S^{\mathcal{L}_\ell}(c(f)), \mathbb{E}_D^{\mathcal{L}_\ell}(c(f))\right) .$$

The other steps of the proof stay exactly the same as the proof of Theorem 18. ■

Appendix B presents more general versions of the last two results.

Let us specialize Theorem 31 to the case where $\mathcal{D}(q,p) = \mathrm{kl}(q\|p)$. Doing so, we recover the classical PAC-Bayesian theorem (Theorem 20), but for aligned posteriors, which therefore has no KL term.

Corollary 32 *For any distribution D on $\mathcal{X}\times\{-1,1\}$, any prior distribution P on a self-complemented set $\mathcal{H}$ of voters $\mathcal{X} \to [-1,1]$, and any $\delta \in (0,1]$, we have*

$$\Pr_{S\sim D^m}\left(\begin{array}{l}\textit{For all posteriors } Q \textit{ aligned on } P: \\ \mathrm{kl}\big(R_S(G_Q) \,\big\|\, R_D(G_Q)\big) \;\le\; \dfrac{1}{m}\left[\ln \dfrac{\xi(m)}{\delta}\right]\end{array}\right) \ge 1-\delta\,,$$

where $\mathrm{kl}(q\|p)$ and $\xi(m)$ and defined by Equations (21) and (22) respectively.

Proof This result follows from Theorem 31 by choosing $\mathcal{D}(q,p) = \mathrm{kl}(q,p)$ and $m' = m$. The rest of the proof relies on Lemma 19 (as for the proof of Theorem 20). ■

The following corollary is very similar to the original PAC-Bayesian bound of McAllester (2003a), but without the KL term.

Corollary 33 *For any distribution D on $\mathcal{X}\times\{-1,1\}$, any self-complemented set $\mathcal{H}$ of voters $\mathcal{X} \to [-1,1]$, any prior distribution P on $\mathcal{H}$, and any $\delta \in (0,1]$, we have*

$$\Pr_{S\sim D^m}\left(\begin{array}{l}\textit{For all posteriors } Q \textit{ aligned on } P: \\ R_D(G_Q) \;\le\; R_S(G_Q) + \sqrt{\dfrac{1}{2m}\left[\ln \frac{\xi(m)}{\delta}\right]}\end{array}\right) \ge 1-\delta\,.$$

Proof The result is derived from Corollary 32, by using $2(q-p)^2 \le \mathrm{kl}(q\|p)$ (Pinsker's inequality), and isolating $R_D(G_Q)$ in the obtained inequality. ■

Unlike Theorem 18, Theorem 31 cannot straightforwardly be used for pairs of voters, as we did in the proof of Theorem 25. The reason is that a posterior distribution that is the result of the product of two aligned posteriors is not necessarily aligned itself. So, we have to ensure that we can get rid of the KL term even in that case.

6.3 PAC-Bayesian Theorems without KL for the Expected Disagreement d_Q^D

The following theorem is similar to Theorem 31 for aligned posteriors, but deals with paired-voters. Instead of the linear loss $\mathcal{L}_\ell$, we use the loss $\mathcal{L}_d$ of Equation (27), which is a linear loss defined on a pair of voters. Again, the next two results can be seen as a particular case of the two theorems from Appendix B.

In this subsection, we use the following shorthand notation. Given $f_{ij} = \langle f_i, f_j \rangle$ as defined in Definition 24, the voters $f_{i^c j}$, f_{ij^c} and $f_{i^c j^c}$ are defined as

$$f_{i^c j}(x) \stackrel{\text{def}}{=} \langle c(f_i)(x), f_j(x) \rangle, \; f_{ij^c}(x) \stackrel{\text{def}}{=} \langle f_i(x), c(f_j)(x) \rangle, \text{ and } f_{i^c j^c}(x) \stackrel{\text{def}}{=} \langle c(f_i)(x), c(f_j)(x) \rangle.$$

Recall that from Equation (26), we have $\mathcal{H}^2 \stackrel{\text{def}}{=} \{f_{ij} : f_i, f_j \in \mathcal{H}\}$ and $Q^2(f_{ij}) \stackrel{\text{def}}{=} Q(f_i) \cdot Q(f_j)$. Similarly, we define $P^2(f_{ij}) \stackrel{\text{def}}{=} P(f_i) \cdot P(f_j)$. Using this notation, let us first generalize the change of measure inequality of Lemma 30 to paired-voters.

Lemma 34 (Change of measure inequality for paired-voters and aligned posteriors) *For any self-complemented set $\mathcal{H}$, for any distribution P on $\mathcal{H}$, any distribution Q aligned on P, and for any measurable function $\phi : \mathcal{H}^2 \to \mathbb{R}$ such that $\phi(f_{ij}) = \phi(f_{i^c j}) = \phi(f_{ij^c}) = \phi(f_{i^c j^c})$ for all $f_{ij} \in \mathcal{H}^2$, we have*

$$\mathop{\mathbf{E}}_{f_{ij} \sim Q^2} \phi(f_{ij}) \;\leq\; \ln \left(\mathop{\mathbf{E}}_{f_{ij} \sim P^2} e^{\phi(f_{ij})} \right).$$

Proof First, note that one can change the expectation over Q^2 to an expectation over P^2, using the fact that $\phi(f_{ij}) = \phi(f_{i^c j}) = \phi(f_{ij^c}) = \phi(f_{i^c j^c})$ for any $f_{ij} \in \mathcal{H}^2$, and that Q is aligned on P. More specifically, we have the following.

$$\begin{aligned}
&4 \cdot \mathop{\mathbf{E}}_{f_{ij} \sim Q^2} \phi(f_{ij}) \\
&= \int_{\mathcal{H}^2} df_{ij} Q^2(f_{ij})\, \phi(f_{ij}) + \int_{\mathcal{H}^2} df_{ij} Q^2(f_{i^c j})\, \phi(f_{i^c j}) + \int_{\mathcal{H}^2} df_{ij} Q^2(f_{ij^c})\, \phi(f_{ij^c}) + \int_{\mathcal{H}^2} df_{ij} Q^2(f_{i^c j^c})\, \phi(f_{i^c j^c}) \\
&= \int_{\mathcal{H}^2} df_{ij}\, Q^2(f_{ij})\, \phi(f_{ij}) + \int_{\mathcal{H}^2} df_{ij}\, Q^2(f_{i^c j})\, \phi(f_{ij}) + \int_{\mathcal{H}^2} df_{ij}\, Q^2(f_{ij^c})\, \phi(f_{ij}) + \int_{\mathcal{H}^2} df_{ij}\, Q^2(f_{i^c j^c})\, \phi(f_{ij}) \\
&= \int_{\mathcal{H}^2} df_{ij} \Big(Q^2(f_{ij}) + Q^2(f_{i^c j}) + Q^2(f_{ij^c}) + Q^2(f_{i^c j^c}) \Big) \phi(f_{ij}) \\
&= \int_{\mathcal{H}^2} df_{ij} \Big(P^2(f_{ij}) + P^2(f_{i^c j}) + P^2(f_{ij^c}) + P^2(f_{i^c j^c}) \Big) \phi(f_{ij}) \\
&\;\;\vdots \\
&= 4 \cdot \mathop{\mathbf{E}}_{f_{ij} \sim P^2} \phi(f_{ij}).
\end{aligned}$$

The result is then obtained by changing the expectation over Q^2 to an expectation over P^2, and then by applying Jensen's inequality (Lemma 47, in Appendix A).

$$\mathop{\mathbf{E}}_{f_{ij}\sim Q^2} \phi(f_{ij}) \;=\; \mathop{\mathbf{E}}_{f_{ij}\sim P^2} \phi(f_{ij}) \;=\; \mathop{\mathbf{E}}_{f_{ij}\sim P^2} \ln e^{\phi(f_{ij})} \;\leq\; \ln\left(\mathop{\mathbf{E}}_{f_{ij}\sim P^2} e^{\phi(f_{ij})}\right).$$

■

Theorem 35 (PAC-Bayesian theorem for paired-voters and aligned posteriors) *For any distribution D on $\mathcal{X}\times\{-1,1\}$, any self-complemented set $\mathcal{H}$ of voters $\mathcal{X}\to[-1,1]$, any prior distribution P on $\mathcal{H}$, any convex function $\mathcal{D} : [0,1]\times[0,1]\to\mathbb{R}$ for which $\mathcal{D}(q,p)=\mathcal{D}(1-q,1-p)$, for any $m'>0$ and any $\delta\in(0,1]$, we have*

$$\Pr_{S\sim D^m}\left(\begin{array}{l}\text{For all posteriors } Q \text{ aligned on } P:\\ \mathcal{D}\big(\,d^S_Q,\,d^D_Q\,\big) \;\leq\; \dfrac{1}{m'}\left[\ln\left(\dfrac{1}{\delta}\mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{f_{ij}\sim P^2} e^{\,m'\cdot\mathcal{D}\left(\mathbb{E}^{\mathcal{L}_d}_S(f_{ij}),\,\mathbb{E}^{\mathcal{L}_d}_D(f_{ij})\right)}\right)\right]\end{array}\right)\geq 1-\delta\,,$$

where f_{ij} is given in Definition 24, and where $P^2(f_{ij})\stackrel{\text{def}}{=}P(f_i)\cdot P(f_j)$.

Proof Theorem 35 is deduced from Theorem 31, by using the change of measure inequality given by Lemma 34 instead of the one from Lemma 30, with $\phi(f_{ij})=m'\cdot\mathcal{D}(\mathbb{E}^{\mathcal{L}_d}_S(f_{ij}),\,\mathbb{E}^{\mathcal{L}_d}_D(f_{ij}))$. As the loss $\mathcal{L}_d$ is such that

$$\mathbb{E}^{\mathcal{L}_d}_{D'}(f_{i^c j^c}) \;=\; \mathbb{E}^{\mathcal{L}_d}_{D'}(f_{ij})\,,\quad\text{ and }\quad \mathbb{E}^{\mathcal{L}_d}_{D'}(f_{i^c j}) \;=\; \mathbb{E}^{\mathcal{L}_d}_{D'}(f_{ij^c}) \;=\; 1-\mathbb{E}^{\mathcal{L}_d}_{D'}(f_{ij})\,,$$

we then have that $\phi(f_{ij})$ has the required property to apply Lemma 34. ■

Let us now specialize Theorem 35 to $\mathcal{D}(q,p)=\mathrm{kl}(q\|p)$.

Corollary 36 *For any distribution D on $\mathcal{X}\times\{-1,1\}$, any self-complemented set $\mathcal{H}$ of voters $\mathcal{X}\to[-1,1]$, any prior distribution P on $\mathcal{H}$, and any $\delta\in(0,1]$, we have*

$$\Pr_{S\sim D^m}\left(\begin{array}{l}\text{For all posteriors } Q \text{ aligned on } P:\\ \mathrm{kl}\big(d^S_Q\,\|\,d^D_Q\big) \;\leq\; \dfrac{1}{m}\left[\ln\frac{\xi(m)}{\delta}\right]\end{array}\right)\geq 1-\delta\,.$$

Proof The result is directly obtained from Theorem 35, by choosing $\mathcal{D}(q,p)=\mathrm{kl}(q,p)$. The rest of the proof relies on Lemma 19. ■

Similarly as for Corollary 33, we can easily derive the following result.

Corollary 37 *For any distribution D on $\mathcal{X}\times\{-1,1\}$, for any self-complemented set $\mathcal{H}$ of voters $\mathcal{X}\to[-1,1]$, any prior distribution P on $\mathcal{H}$, and any $\delta\in(0,1]$, we have*

$$\Pr_{S\sim D^m}\left(\begin{array}{l}\text{For all posteriors } Q \text{ aligned on } P:\\ d^D_Q \;\geq\; d^S_Q-\sqrt{\dfrac{1}{2m}\left[\ln\frac{\xi(m)}{\delta}\right]}\end{array}\right)\geq 1-\delta\,.$$

Proof The result is derived from Corollary 36, by using $2(q-p)^2 \leq \mathrm{kl}(q\|p)$ (Pinsker's inequality), and isolating d_Q^D in the obtained inequality. ■

6.4 A Bound for the Risk of the Majority Vote without KL Term

Finally, we make use of these results to bound $\mathcal{C}_Q^D$ – and therefore $R_D(B_Q)$ – for aligned posteriors Q, giving rise to PAC-Bound 3. Aside from the fact that this bound has no KL term, it is similar to PAC-Bound 1, as it separately bounds the Gibbs risk and the expected disagreement. This new PAC-Bayesian bound provides us with a starting point to design the MinCq leaning algorithm introduced in Section 8.

PAC-Bound 3 *For any distribution D on $\mathcal{X}\times\{-1,1\}$, for any self-complemented set $\mathcal{H}$ of voters $\mathcal{X}\to[-1,1]$, for any prior distribution P on $\mathcal{H}$, and any $\delta\in(0,1]$, we have*

$$\Pr_{S\sim D^m}\left(\begin{array}{l}\forall\, Q \text{ aligned on } P:\\ R_D(B_Q) \;\leq\; 1-\dfrac{\left(1-2\cdot\overline{r}\right)^2}{1-2\cdot\underline{d}} \;=\; 1-\dfrac{\left(\underline{\mu_1}\right)^2}{\overline{\mu_2}}\end{array}\right) \geq 1-\delta\,,$$

where

$$\overline{r} \stackrel{\text{def}}{=} \min\left(\tfrac{1}{2},\, R_S(G_Q)+\sqrt{\tfrac{1}{2m}\left[\ln\tfrac{\xi(m)}{\delta/2}\right]}\right), \qquad \underline{d} \stackrel{\text{def}}{=} \max\left(0,\, d_Q^S-\sqrt{\tfrac{1}{2m}\left[\ln\tfrac{\xi(m)}{\delta/2}\right]}\right),$$

$$\underline{\mu_1} \stackrel{\text{def}}{=} \max\left(0,\, \mu_1(M_Q^S)-\sqrt{\tfrac{2}{m}\left[\ln\tfrac{\xi(m)}{\delta/2}\right]}\right), \qquad \overline{\mu_2} \stackrel{\text{def}}{=} \min\left(1,\, \mu_2(M_Q^S)+\sqrt{\tfrac{2}{m}\left[\ln\tfrac{\xi(m)}{\delta/2}\right]}\right).$$

Proof The inequality is a consequence of Theorem 11, as well as Corollaries 33 and 37. The equality $1-\frac{(1-2\cdot\overline{r})^2}{1-2\cdot\underline{d}} = 1-\frac{(\underline{\mu_1})^2}{\overline{\mu_2}}$ is a direct application of Equations (7) and (9). ■

PAC-Bound 3' that is presented at the end of Section 7 accepts voters that are kernel functions defined using a part of the training set S. This is unusual in the PAC-Bayesian theory, since the prior P on the set of voters has to be defined before seeing the training set S. To overcome this difficulty, we use the sample compression theory.

7. PAC-Bayesian Theory for Sample-Compressed Voters

PAC-Bayesian theorems of Sections 5 and 6 are not valid when $\mathcal{H}$ consists of a set of functions of the form $\pm k(x_i,\cdot)$ for some kernel $k:\mathcal{X}\times\mathcal{X}\to[-1,1]$, as is the case with the Support Vector Machine classifier (see Equation 1). This is because the definition of each involved voter depends on an example (x_i,y_i) of the training data S. This is problematic from the PAC-Bayesian point of view because the prior on the voters is supposed to be defined before seeing the data S. There are two known methods to overcome this problem.

The first method, introduced by Langford and Shawe-Taylor (2002), considers a surrogate set of voters $\mathcal{H}^k$ of *all* the linear classifiers in the space induced[12] by the kernel k. They

12. This space is also known as a Reproducible Kernel Hilbert Space (RKHS). For more details, see Cristianini and Shawe-Taylor (2000) and Schölkopf et al. (2001)

then make use of the representer theorem to show that the classification function turns out to be a linear combination of the examples, similar to the Support Vector Machine classifier (Equation 1). To avoid the curse of dimensionality, they propose restricting the choice of the prior and posterior distributions on $\mathcal{H}^k$ to isotropic Gaussian centered on a vector representing a particular linear classifier. Based on this approach, Germain et al. (2009) suggests a learning algorithm for linear classifiers that exactly consists in a PAC-Bayesian bound minimization.

The second method, that is presented in the present section, is based on the sample compression setting of Floyd and Warmuth (1995). It has been adapted to the PAC-Bayesian theory by Laviolette and Marchand (2005, 2007), allowing one to directly deal with the case where voters are constructed using examples in the training set, without involving any RKHS notion nor any representer theorem. Conversely to the first method described above, the sample compression approach allows one not only to deal with kernel functions, but with any kind of similarity measure between examples, hence to deal with any kind of voters.

7.1 The General Sample Compression Setting

In the *sample compression setting*, learning algorithms have access to a data-dependent set of voters, that we refer to as *sc-voters*. Given a training sequence[13] $S = \langle\, (x_1, y_1), \ldots, (x_m, y_m) \,\rangle$, each sc-voter is described by a sequence $S_{\mathbf{i}}$ of elements of S called the *compression sequence*, and a *message* σ which represents the additional information needed to obtain a voter from $S_{\mathbf{i}}$. If $\mathbf{i} = \langle i_1, i_2, .., i_k \rangle$, then $S_{\mathbf{i}} \stackrel{\text{def}}{=} \langle (x_{i_1}, y_{i_1}), (x_{i_2}, y_{i_2}), \ldots, (x_{i_k}, y_{i_k}) \rangle$. In this paper, repetitions are allowed in $S_{\mathbf{i}}$, and k, the number of indices present in $\mathbf{i}$ (counting the repetitions), is denoted by $|\mathbf{i}|$.

The fact that each sc-voter is described by a compression sequence and a message implies that there exists a *reconstruction function* $\mathcal{R}(S_{\mathbf{i}}, \sigma)$ that outputs a classifier when given an arbitrary compression sequence $S_{\mathbf{i}}$ and a message σ. The message σ is chosen from the set $\Sigma_{S_{\mathbf{i}}}$ of all messages that can be supplied with the compression sequence $S_{\mathbf{i}}$. In the PAC-Bayesian setting, $\Sigma_{S_{\mathbf{i}}}$ must be defined a priori (before observing the training data) for all possible sequences $S_{\mathbf{i}}$, and can be either a discrete or a continuous set. The sample compression setting strictly generalizes the (classical) non-sample-compressed setting, since the latter corresponds to the case where $|\mathbf{i}| = 0$, the voters being then defined only via the messages.

7.2 A Simplified Sample Compression Setting

For the needs of this paper, we consider a simplified framework where sc-voters have a compression sequence of at most λ examples (possibly with repetitions) and a message string of λ bits that we represent by a sequence of "-1" and "$+1$". Instead of being defined on sc-voters, the weighted distribution Q is defined on $\mathcal{I}_\lambda \times \Sigma_\lambda$, where

$$\mathcal{I}_\lambda \stackrel{\text{def}}{=} \Big\{ \langle i_1, i_2, .., i_k \rangle \,:\, k \in \{0, .., \lambda\} \text{ and } i_j \in \{1, .., m\} \Big\} \quad \text{and} \quad \Sigma_\lambda \stackrel{\text{def}}{=} \Big\{ -1, 1 \Big\}^\lambda . \quad (38)$$

13. The sample compression theory considers the training examples as a sequence instead of a set, because it refers to the training examples by their indices.

In other words, $Q(\mathbf{i}, \boldsymbol{\sigma})$ corresponds to the weight of the sc-voter output by $\mathcal{R}(S_{\mathbf{i}}, \boldsymbol{\sigma})$, *i.e.*, the sc-voter of compression sequence $\mathbf{i} = \langle i_1, \ldots, i_{|\mathbf{i}|} \rangle \in \mathcal{I}_\lambda$ and message $\boldsymbol{\sigma} = \langle \sigma_1, \ldots, \sigma_\lambda \rangle \in \Sigma_\lambda$. In particular, a prior (resp., a posterior) on the set of all sc-voters is now simply a prior on the set $\mathcal{I}_\lambda \times \Sigma_\lambda$. Thus, such a prior can really be defined *a priori*, before seeing the data S.[14] The set of sc-voters is therefore only defined when the training sequence S is given, and corresponds to

$$\mathcal{H}^{\mathcal{R}}_{S,\lambda} \stackrel{\text{def}}{=} \{\mathcal{R}(S_{\mathbf{i}}, \boldsymbol{\sigma}) \,:\, \mathbf{i} \in \mathcal{I}_\lambda,\, \boldsymbol{\sigma} \in \Sigma_\lambda\}\,.$$

Finally, given a training sequence S and a reconstruction function $\mathcal{R}$, for a distribution Q on $\mathcal{I}_\lambda \times \Sigma_\lambda$, we define the Bayes classifier as

$$B_{Q,S} \stackrel{\text{def}}{=} \operatorname{sgn}\left[\mathop{\mathbf{E}}_{(\mathbf{i},\boldsymbol{\sigma})\sim Q} \mathcal{R}(S_{\mathbf{i}}, \boldsymbol{\sigma})\right].$$

We then define the Bayes risk $R_{D'}(B_{Q,S})$ and the Gibbs risk $R_{D'}(G_{Q,S})$ of a distribution Q on $\mathcal{I}_\lambda \times \Sigma_\lambda$ relative to D' as

$$\begin{aligned} R_{D'}(B_{Q,S}) &\stackrel{\text{def}}{=} \mathbb{E}^{\mathcal{L}_{01}}_{D'}\big(B_{Q,S}\big)\,, \\ R_{D'}(G_{Q,S}) &\stackrel{\text{def}}{=} \mathop{\mathbf{E}}_{(\mathbf{i},\boldsymbol{\sigma})\sim Q} \mathbb{E}^{\mathcal{L}_\ell}_{D'}\big(\mathcal{R}(S_{\mathbf{i}}, \boldsymbol{\sigma})\big)\,. \end{aligned}$$

7.3 A First Sample-Compressed PAC-Bayesian Theorem

To derive PAC-Bayesian bounds for majority votes of sc-voters, one must deal with the following issue: even if the training sequence S is drawn i.i.d. from a data-generating distribution D, the empirical risk of the Gibbs $R_S(G_{Q,S})$ is not an unbiased estimate of its true risk $R_D(G_{Q,S})$. For instance, the reconstruction function $\mathcal{R}$ can be such that an sc-voter output by $\mathcal{R}(S_{\mathbf{i}}, \boldsymbol{\sigma})$ never errs on an example belonging to its compression sequence $S_{\mathbf{i}}$; this biases the empirical risk because examples of $S_{\mathbf{i}}$ are all in S.

To deal with this bias, the $\frac{1}{m}$ factor in the usual PAC-Bayesian bounds is replaced by a factor of the form $\frac{1}{m-l}$ in their sample compression versions. In Laviolette and Marchand (2005, 2007), l corresponds to the Q-average size of the sample compression sequence. In the present paper, we restrain ourselves to a simpler case, where l is the maximum possible size of a compression sequence (*i.e.*, $l = \lambda$). This simplification allows us to deal with the biased character of the empirical Gibbs risk using a proof approach similar to the one proposed in Germain et al. (2011). The key step of this approach is summarized in the following lemma.

Lemma 38 *Let $\mathcal{R}$ be a reconstruction function that outputs sc-voters of size at most λ (where $\lambda < m$). For any distribution D on $\mathcal{X} \times \{-1, 1\}$, and for any prior distribution P on $\mathcal{I}_\lambda \times \Sigma_\lambda$,*

$$\mathop{\mathbf{E}}_{S\sim D^m} \mathop{\mathbf{E}}_{(\mathbf{i},\boldsymbol{\sigma})\sim P} e^{(m-\lambda)\cdot 2\cdot\left(\mathbb{E}^{\mathcal{L}_\ell}_S(\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma})) - \mathbb{E}^{\mathcal{L}_\ell}_D(\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma}))\right)^2} \;\leq\; e^{4\,\lambda} \cdot \xi(m-\lambda)\,,$$

where $\xi(\cdot)$ is defined by Equation (22), and therefore we have that $\xi(m-\lambda) \leq 2\sqrt{m-\lambda}$.

14. Laviolette and Marchand (2007) describe a more general setting where, for each $S \in (\mathcal{X} \times \mathcal{Y})^m$, a prior is defined on $\mathcal{I}_\lambda \times \Sigma_{S_{\mathbf{i}}}$. Hence, the messages may depend on the compression sequence $S_{\mathbf{i}}$.

Proof As the the choice of $(\mathbf{i},\boldsymbol{\sigma})$ according to the prior P is independent[15] of S, we have

$$\begin{aligned}
&\mathop{\mathbf{E}}_{S\sim D^m}\ \mathop{\mathbf{E}}_{(\mathbf{i},\boldsymbol{\sigma})\sim P} e^{(m-\lambda)\cdot 2\cdot\left(\mathbb{E}_S^{\mathcal{L}_\ell}(\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma}))-\mathbb{E}_D^{\mathcal{L}_\ell}(\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma}))\right)^2}\\
&\quad= \mathop{\mathbf{E}}_{(\mathbf{i},\boldsymbol{\sigma})\sim P}\ \mathop{\mathbf{E}}_{S\sim D^m} e^{(m-\lambda)\cdot 2\cdot\left(\mathbb{E}_S^{\mathcal{L}_\ell}(\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma}))-\mathbb{E}_D^{\mathcal{L}_\ell}(\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma}))\right)^2} \qquad (39)\\
&\quad= \mathop{\mathbf{E}}_{(\mathbf{i},\boldsymbol{\sigma})\sim P}\ \mathop{\mathbf{E}}_{S_{\mathbf{i}}\sim D^\lambda}\ \mathop{\mathbf{E}}_{S_{\mathbf{i}^c}\sim D^{m-\lambda}} e^{(m-\lambda)\cdot 2\cdot\left(\mathbb{E}_S^{\mathcal{L}_\ell}(\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma}))-\mathbb{E}_D^{\mathcal{L}_\ell}(\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma}))\right)^2}. \qquad (40)
\end{aligned}$$

Let us now rewrite the empirical loss of an sc-voter as a combination of the loss on its compression sequence $S_{\mathbf{i}}$ and the loss on the other training examples $S_{\mathbf{i}^c}$.

$$\mathbb{E}_S^{\mathcal{L}_\ell}(\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma})) \;=\; \frac{1}{m}\left[\lambda\cdot\mathbb{E}_{S_{\mathbf{i}}}^{\mathcal{L}_\ell}(\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma}))+(m-\lambda)\cdot\mathbb{E}_{S_{\mathbf{i}^c}}^{\mathcal{L}_\ell}(\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma}))\right].$$

Since $0\le\mathbb{E}_{D'}^{\mathcal{L}_\ell}(\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma}))\le 1$ and $2\cdot(q-p)^2\le \mathrm{kl}(q\|p)$ (Pinsker's inequality), we have

$$\begin{aligned}
&(m-\lambda)\cdot 2\cdot\left(\mathbb{E}_S^{\mathcal{L}_\ell}(\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma}))-\mathbb{E}_D^{\mathcal{L}_\ell}(\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma}))\right)^2\\
&\quad= (m-\lambda)\cdot 2\cdot\left(\tfrac{1}{m}\left[\lambda\cdot\mathbb{E}_{S_{\mathbf{i}}}^{\mathcal{L}_\ell}(\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma}))+(m-\lambda)\cdot\mathbb{E}_{S_{\mathbf{i}^c}}^{\mathcal{L}_\ell}(\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma}))\right]-\mathbb{E}_D^{\mathcal{L}_\ell}(\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma}))\right)^2\\
&\quad= (m-\lambda)\cdot 2\cdot\left(\tfrac{\lambda}{m}\left[\mathbb{E}_{S_{\mathbf{i}}}^{\mathcal{L}_\ell}(\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma}))-\mathbb{E}_{S_{\mathbf{i}^c}}^{\mathcal{L}_\ell}(\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma}))\right]+\left[\mathbb{E}_{S_{\mathbf{i}^c}}^{\mathcal{L}_\ell}(\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma}))-\mathbb{E}_D^{\mathcal{L}_\ell}(\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma}))\right]\right)^2\\
&\quad= (m-\lambda)\cdot 2\cdot\left(\left(\tfrac{\lambda}{m}\right)^2\left[\mathbb{E}_{S_{\mathbf{i}}}^{\mathcal{L}_\ell}(\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma}))-\mathbb{E}_{S_{\mathbf{i}^c}}^{\mathcal{L}_\ell}(\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma}))\right]^2+\left[\mathbb{E}_{S_{\mathbf{i}^c}}^{\mathcal{L}_\ell}(\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma}))-\mathbb{E}_D^{\mathcal{L}_\ell}(\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma}))\right]^2\right.\\
&\qquad\qquad\left.+\tfrac{2\lambda}{m}\left[\mathbb{E}_{S_{\mathbf{i}}}^{\mathcal{L}_\ell}(\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma}))-\mathbb{E}_{S_{\mathbf{i}^c}}^{\mathcal{L}_\ell}(\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma}))\right]\left[\mathbb{E}_{S_{\mathbf{i}^c}}^{\mathcal{L}_\ell}(\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma}))-\mathbb{E}_D^{\mathcal{L}_\ell}(\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma}))\right]\right)\\
&\quad\le (m-\lambda)\cdot 2\cdot\left(\left(\tfrac{\lambda}{m}\right)^2+\left[\mathbb{E}_{S_{\mathbf{i}^c}}^{\mathcal{L}_\ell}(\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma}))-\mathbb{E}_D^{\mathcal{L}_\ell}(\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma}))\right]^2+\tfrac{2\lambda}{m}\right)\\
&\quad= 2\,\lambda\cdot\left(2-\tfrac{\lambda}{m}-\left(\tfrac{\lambda}{m}\right)^2\right)+(m-\lambda)\cdot 2\cdot\left[\mathbb{E}_{S_{\mathbf{i}^c}}^{\mathcal{L}_\ell}(\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma}))-\mathbb{E}_D^{\mathcal{L}_\ell}(\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma}))\right]^2\\
&\quad\le 4\,\lambda+(m-\lambda)\cdot 2\cdot\left[\mathbb{E}_{S_{\mathbf{i}^c}}^{\mathcal{L}_\ell}(\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma}))-\mathbb{E}_D^{\mathcal{L}_\ell}(\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma}))\right]^2\\
&\quad\le 4\,\lambda+(m-\lambda)\cdot\mathrm{kl}\left(\mathbb{E}_{S_{\mathbf{i}^c}}^{\mathcal{L}_\ell}(\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma}))\,\|\,\mathbb{E}_D^{\mathcal{L}_\ell}(\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma}))\right). \qquad (41)
\end{aligned}$$

Note that $\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma})$ does not depend on examples contained in $S_{\mathbf{i}^c}$. Thus, from the point of view of $S_{\mathbf{i}^c}$, $\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma})$ is a classical voter (not a sample-compressed one). Therefore, one can apply Lemma 19, replacing $S\sim D^m$ by $S_{\mathbf{i}^c}\sim D^{m-\lambda}$, and f by $\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma})$. Lemma 19, together with Equations (40) and (41), gives

$$\begin{aligned}
&\mathop{\mathbf{E}}_{(\mathbf{i},\boldsymbol{\sigma})\sim P}\ \mathop{\mathbf{E}}_{S_{\mathbf{i}}\sim D^\lambda}\ \mathop{\mathbf{E}}_{S_{\mathbf{i}^c}\sim D^{m-\lambda}} e^{(m-\lambda)\cdot 2\cdot\left(\mathbb{E}_S^{\mathcal{L}_\ell}(\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma}))-\mathbb{E}_D^{\mathcal{L}_\ell}(\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma}))\right)^2}\\
&\qquad\le e^{4\,\lambda}\cdot\mathop{\mathbf{E}}_{(\mathbf{i},\boldsymbol{\sigma})\sim P}\ \mathop{\mathbf{E}}_{S_{\mathbf{i}}\sim D^\lambda}\ \mathop{\mathbf{E}}_{S_{\mathbf{i}^c}\sim D^{m-\lambda}} e^{(m-\lambda)\cdot\mathrm{kl}\left(\mathbb{E}_{S_{\mathbf{i}^c}}^{\mathcal{L}_\ell}(\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma}))\,\|\,\mathbb{E}_D^{\mathcal{L}_\ell}(\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma}))\right)}\\
&\qquad\le e^{4\,\lambda}\cdot\mathop{\mathbf{E}}_{(\mathbf{i},\boldsymbol{\sigma})\sim P}\ \mathop{\mathbf{E}}_{S_{\mathbf{i}}\sim D^\lambda}\ \xi(m-\lambda)\;=\;e^{4\,\lambda}\cdot\xi(m-\lambda),
\end{aligned}$$

and we are done. ■

15. Note that because of this independence, the exchange in the order of the two expectations (Line 39) is trivial. This independence is a direct consequence of our choice to only consider the simplified setting described by Equation (38). In the more general setting of Laviolette and Marchand (2007), this part of the proof is more complicated.

The next PAC-Bayesian theorem presents the generalization of McAllester's PAC-Bayesian bound (Corollary 22) for the sample compression case.

Theorem 39 *Let $\mathcal{R}$ be a reconstruction function that outputs sc-voters of size at most λ (where $\lambda < m$). For any distribution D on $\mathcal{X}\times\{-1,1\}$, for any prior distribution P on $\mathcal{I}_\lambda \times \Sigma_\lambda$, and any $\delta \in (0,1]$, we have*

$$\Pr_{S\sim D^m}\left(\begin{array}{l} \textit{For all posteriors } Q \textit{ :} \\ R_D(G_{Q,S}) \;\leq\; R_S(G_{Q,S}) + \sqrt{\frac{1}{2(m-\lambda)}\left[\mathrm{KL}(Q\|P) + 4\lambda + \ln\frac{\xi(m-\lambda)}{\delta}\right]} \end{array}\right) \geq 1-\delta\,.$$

Proof We apply the exact same steps as in the proof of Theorem 18, with $m' = m-\lambda$, $f = \mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma})$, and $\mathcal{D}(q,p) = 2(q-p)^2$, we obtain

$$\Pr_{S\sim D^m}\left(\begin{array}{l} \text{For all posteriors } Q \text{ :} \\ 2\Big(R_S(G_{Q,S}) - R_D(G_{Q,S})\Big)^2 \\ \quad\leq \frac{1}{m-\lambda}\left[\mathrm{KL}(Q\|P) + \ln\left(\frac{1}{\delta}\mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{(\mathbf{i},\boldsymbol{\sigma})\sim P} e^{(m-\lambda)\cdot 2\cdot\left(\mathbb{E}_S^{\mathcal{L}_\ell}(\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma})) - \mathbb{E}_D^{\mathcal{L}_\ell}(\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma}))\right)^2}\right)\right] \end{array}\right) \geq 1-\delta\,.$$

The result then follows from Lemma 38 and easy calculations. ■

All the PAC-Bayesian results presented in the preceding sections can be similarly generalized. We leave them to the reader with the exception of the PAC-Bayesian bounds that have no KL, that are used in the next section, as we present the learning algorithm MinCq that minimizes the $\mathcal{C}$-bound.

7.4 Sample-Compressed PAC-Bayesian Bounds without KL

The bounds presented in this section generalize the results presented in Section 6 to the sample compression case. We first need to generalize the notion of self-complement (Definition 29) to sc-voters.

Definition 40 A reconstruction function $\mathcal{R}$ is said to be *self-complemented* if for any training sequence $S \in (\mathcal{X}\times\mathcal{Y})^m$ and any $(\mathbf{i},\boldsymbol{\sigma}) \in \mathcal{I}_\lambda \times \Sigma_\lambda$, we have

$$-\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma}) \;\;=\;\; \mathcal{R}(S_{\mathbf{i}},-\boldsymbol{\sigma})\,,$$

where, if $\boldsymbol{\sigma} = \langle\sigma_1,..,\sigma_\lambda\rangle$, then $-\boldsymbol{\sigma} = \langle-\sigma_1,..,-\sigma_\lambda\rangle$.

7.4.1 A PAC-Bayesian Theorem for the Gibbs Risk of Sc-Voters

Theorem 41 *Let $\mathcal{R}$ be a self-complemented reconstruction function that outputs sc-voters of size at most λ (where $\lambda < m$). For any distribution D on $\mathcal{X}\times\{-1,1\}$, for any prior distribution P on $\mathcal{I}_\lambda \times \Sigma_\lambda$, and any $\delta \in (0,1]$, we have*

$$\Pr_{S\sim D^m}\left(\begin{array}{l} \textit{For all posteriors } Q \textit{ aligned on } P \textit{ :} \\ R_D(G_{Q,S}) \;\leq\; R_S(G_{Q,S}) + \sqrt{\frac{1}{2(m-\lambda)}\left[4\lambda + \ln\frac{\xi(m-\lambda)}{\delta}\right]} \end{array}\right) \geq 1-\delta\,.$$

Proof First note that $2\cdot(q-p)^2 = 2\cdot((1-q)-(1-p))^2$. Then apply the exact same steps as in the proof of Theorem 31 with $m' = m-\lambda$, $f = \mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma})$, and $\mathcal{D}(q,p) = 2(q-p)^2$ to obtain

$$\Pr_{S\sim D^m}\left(\begin{array}{l}\text{For all posteriors } Q \text{ aligned on } P\text{:}\\ 2\Big(R_S(G_{Q,S})-R_D(G_{Q,S})\Big)^2 \le \dfrac{1}{m-\lambda}\left[\ln\left(\dfrac{1}{\delta}\mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{(\mathbf{i},\boldsymbol{\sigma})\sim P} e^{(m-\lambda)\cdot 2\cdot\left(\mathbb{E}_S^{\mathcal{L}_\ell}(\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma}))-\mathbb{E}_D^{\mathcal{L}_\ell}(\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma}))\right)^2}\right)\right]\end{array}\right) \ge 1-\delta\,.$$

The result then follows from Lemma 38 and easy calculations. ■

7.4.2 A PAC-Bayesian Theorem for the Disagreement of Sc-Voters

Given a training sequence S and a reconstruction function $\mathcal{R}$, we define the expected disagreement of a distribution Q on $\mathcal{I}_\lambda\times\Sigma_\lambda$ relative to D' as

$$\begin{aligned} d^{D'}_{Q,S} &\stackrel{\text{def}}{=} \mathop{\mathbf{E}}_{x\sim D'_{\mathcal{X}}}\ \mathop{\mathbf{E}}_{(\mathbf{i},\boldsymbol{\sigma})\sim Q}\ \mathop{\mathbf{E}}_{(\mathbf{i}',\boldsymbol{\sigma}')\sim Q} \mathcal{L}_\ell\left(\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma})(x),\mathcal{R}(S_{\mathbf{i}'},\boldsymbol{\sigma}')(x)\right)\\ &= \mathop{\mathbf{E}}_{(\mathbf{i},\mathbf{i}',\boldsymbol{\sigma},\boldsymbol{\sigma}')\sim Q^2}\mathbb{E}_{D'}^{\mathcal{L}_d}\left(\overline{\mathcal{R}}(S_{\mathbf{i},\mathbf{i}'},\boldsymbol{\sigma},\boldsymbol{\sigma}')\right),\end{aligned}$$

where

$$\begin{aligned} Q^2(\mathbf{i},\mathbf{i}',\boldsymbol{\sigma},\boldsymbol{\sigma}') &\stackrel{\text{def}}{=} Q(\mathbf{i},\boldsymbol{\sigma})\cdot Q(\mathbf{i}',\boldsymbol{\sigma}')\,,\\ \overline{\mathcal{R}}(S_{\mathbf{i},\mathbf{i}'},\boldsymbol{\sigma},\boldsymbol{\sigma}')(x) &\stackrel{\text{def}}{=} \left\langle\,\mathcal{R}(S_{\mathbf{i}},\boldsymbol{\sigma})(x),\mathcal{R}(S_{\mathbf{i}'},\boldsymbol{\sigma}')(x)\,\right\rangle\,.\end{aligned}$$

Thus, $\overline{\mathcal{R}}$ is a new reconstruction function that outputs an *sc-paired-voter* which is the sample-compressed version of the paired-voter of Definition 24. From there, we adapt Corollary 37 to sc-voters, and we obtain the following PAC-Bayesian theorem. This result bounds $d^D_{Q,S}$ for posterior distributions Q aligned on a prior distribution P.

Theorem 42 *Let $\mathcal{R}$ be a self-complemented reconstruction function that outputs sc-voters of size at most λ (where $\lambda < \lfloor\frac{m}{2}\rfloor$). For any distribution D on $\mathcal{X}\times\{-1,1\}$, for any prior distribution P on $\mathcal{I}_\lambda\times\Sigma_\lambda$, and any $\delta\in(0,1]$, we have*

$$\Pr_{S\sim D^m}\left(\begin{array}{l}\textit{For all posteriors } Q \textit{ aligned on } P\text{:}\\ d^D_{Q,S} \ge d^S_{Q,S} - \sqrt{\dfrac{1}{2(m-2\lambda)}\left[8\lambda+\ln\frac{\xi(m-2\lambda)}{\delta}\right]}\end{array}\right) \ge 1-\delta\,.$$

Proof Let $P^2(\mathbf{i},\mathbf{i}',\boldsymbol{\sigma},\boldsymbol{\sigma}') \stackrel{\text{def}}{=} P(\mathbf{i},\boldsymbol{\sigma})\cdot P(\mathbf{i}',\boldsymbol{\sigma}')$. Now note that $2\cdot(q-p)^2 = 2\cdot((1-q)-(1-p))^2$. Then apply the exact same steps as in the proof of Theorem 35 with $m' = m-2\lambda$, $f_{ij} = \overline{\mathcal{R}}(S_{\mathbf{i},\mathbf{i}'},\boldsymbol{\sigma},\boldsymbol{\sigma}')$ and $\mathcal{D}(q,p) = 2(q-p)^2$ to obtain

$$\Pr_{S\sim D^m}\left(\begin{array}{l}\text{For all posteriors } Q \text{ aligned on } P\text{:}\\ 2\Big(d^S_{Q,S}-d^D_{Q,S}\Big)^2 \le \dfrac{1}{m}\left[\ln\left(\dfrac{1}{\delta}\mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{(\mathbf{i},\mathbf{i}',\boldsymbol{\sigma},\boldsymbol{\sigma}')\sim P^2} e^{m\cdot 2\cdot\left(\mathbb{E}_S^{\mathcal{L}_d}(\overline{\mathcal{R}}(S_{\mathbf{i},\mathbf{i}'},\boldsymbol{\sigma},\boldsymbol{\sigma}'))-\mathbb{E}_D^{\mathcal{L}_d}(\overline{\mathcal{R}}(S_{\mathbf{i},\mathbf{i}'},\boldsymbol{\sigma},\boldsymbol{\sigma}'))\right)^2}\right)\right]\end{array}\right) \ge 1-\delta\,.$$

Calculations similar to the ones of the proof of Lemma 38 (with λ replaced by 2λ) give

$$\mathop{\mathbf{E}}_{S\sim D^m}\ \mathop{\mathbf{E}}_{(\mathbf{i},\mathbf{i}',\boldsymbol{\sigma},\boldsymbol{\sigma}')\sim P^2} e^{(m-2\lambda)\cdot 2\cdot\left(\mathbb{E}_S^{\mathcal{L}_d}(\overline{\mathcal{R}}(S_{\mathbf{i},\mathbf{i}'},\boldsymbol{\sigma},\boldsymbol{\sigma}'))-\mathbb{E}_D^{\mathcal{L}_d}(\overline{\mathcal{R}}(S_{\mathbf{i},\mathbf{i}'},\boldsymbol{\sigma},\boldsymbol{\sigma}'))\right)^2} \ \leq\ e^{8\,\lambda}\cdot\xi(m-2\lambda)\,.$$

Therefore, we have

$$\Pr_{S\sim D^m}\left(\begin{array}{l}\text{For all posteriors } Q \text{ aligned on } P\text{:}\\ 2\Big(d^S_{Q,S}-d^D_{Q,S}\Big)^2\leq\dfrac{1}{m-2\lambda}\Big[8\lambda+\ln\frac{\xi(m-2\,\lambda)}{\delta}\Big]\end{array}\right)\ \geq\ 1-\delta\,.$$

and the result is obtained by isolating $d^D_{Q,S}$ in the inequality. ■

7.4.3 A Sample Compression Bound for the Risk of the Majority Vote

Let us now exploit Theorems 41 and 42, together with the $\mathcal{C}$-bound of Theorem 11, to obtain a bound on the risk on a majority vote with kernel functions as voters. Given any similarity function (possibly a kernel) $k : \mathcal{X}\times\mathcal{X}\to[-1,1]$ and a training sequence size of m, we consider a majority vote of sc-voters of compression size at most 1 given by the following reconstruction function,

$$\mathcal{R}_k\big(S_{\mathbf{i}},\langle\sigma\rangle\big)(x)\ \stackrel{\text{def}}{=}\ \begin{cases}\sigma & \text{if } \mathbf{i}=\langle\,\rangle,\\ \sigma\cdot k(x_i,x) & \text{otherwise } (\ \mathbf{i}=\langle i\rangle\ ),\end{cases}$$

where $\mathbf{i}\in\mathcal{I}_1=\{\langle\,\rangle,\langle 1\rangle,\langle 2\rangle,\ldots,\langle m\rangle\}$ and $\langle\sigma\rangle\in\Sigma_1$ (thus, $\sigma\in\{-1,1\}$). Here, the elements of sets $\mathcal{I}_1$ and Σ_1 are obtained from Equation (38), with $\lambda=1$. Note that $\mathcal{R}_k$ is self-complemented (Definition 40) because $-\mathcal{R}_k\big(S_{\mathbf{i}},\langle\sigma\rangle\big)=\mathcal{R}_k\big(S_{\mathbf{i}},\langle-\sigma\rangle\big)$ for any $(\mathbf{i},\boldsymbol{\sigma})$.

Once the training sequence $S\sim D^m$ is observed, the (self-complemented) reconstruction function $\mathcal{R}_k$ gives rise to the following set of $2m+2$ sc-voters,

$$\mathcal{H}^{\mathcal{R}_k}_{S,1}\ \stackrel{\text{def}}{=}\ \Big\{b(\cdot),k(x_1,\cdot),k(x_2,\cdot),\ldots,k(x_m,\cdot),-b(\cdot),-k(x_1,\cdot),-k(x_2,\cdot),\ldots,-k(x_m,\cdot)\Big\}\,,$$

where $b:\mathcal{X}\to\{1\}$ is a "dummy voter" that always outputs 1 and allows introducing a *bias* value into the majority vote classifier. Note that $\mathcal{H}^{\mathcal{R}_k}_{S,1}$ is a self-complemented set of sc-voters, and the margin of the majority vote given by the distribution Q on $\mathcal{H}^{\mathcal{R}_k}_{S,1}$ is

$$M_{Q,S}(x,y)\ \stackrel{\text{def}}{=}\ y\left(Q\big(\,b(\cdot)\,\big)-Q\big(\,-b(\cdot)\,\big)+\sum_{i=1}^m\big[Q\big(\,k(x_i,\cdot)\,\big)-Q\big(\,-k(x_i,\cdot)\,\big)\big]\,k(x_i,x)\right).$$

Consequently, the empirical first and second moments of this margin are

$$\mu_1(M^S_{Q,S})\ =\ \frac{1}{m}\sum_{i=1}^m M_{Q,S}(x_i,y_i),\quad\text{and}\quad\mu_2(M^S_{Q,S})\ =\ \frac{1}{m}\sum_{i=1}^m\Big[M_{Q,S}(x_i,y_i)\Big]^2.$$

Hence, the empirical Gibbs risk and the empirical expected disagreement can be expressed by

$$R_S(G_{Q,S})\ =\ \frac{1}{2}\left(1-\mu_1(M^S_{Q,S})\right),\quad\text{and}\quad d^S_{Q,S}\ =\ \frac{1}{2}\left(1-\mu_2(M^S_{Q,S})\right).\tag{42}$$

Thus, we obtain the following bound on the risk of a majority vote of kernel voters $R_D(B_{Q,S})$ for aligned posteriors Q.

PAC-Bound 3' *Let $k : \mathcal{X} \times \mathcal{X} \to [-1, 1]$. For any distribution D on $\mathcal{X} \times \{-1, 1\}$, for any prior distribution P on $\mathcal{H}_{S,1}^{\mathcal{R}_k}$, and any $\delta \in (0, 1]$, we have*

$$\Pr_{S\sim D^m}\left(\begin{array}{l}\forall\, Q \text{ aligned on } P\ :\\ \\ R_D(B_{Q,S}) \ \leq\ 1-\dfrac{\left(1-2\cdot\overline{r}\right)^2}{1-2\cdot\underline{d}} \ =\ 1-\dfrac{\left(\underline{\mu_1}\right)^2}{\overline{\mu_2}}\end{array}\right) \geq 1-\delta\,,$$

where

$$\begin{aligned}
\overline{r} &\stackrel{\text{def}}{=} \min\left(\tfrac{1}{2},\, R_S(G_{Q,S}) + \sqrt{\tfrac{1}{2(m-1)}\left[4+\ln\tfrac{\xi(m-1)}{\delta/2}\right]}\right),\\
\underline{d} &\stackrel{\text{def}}{=} \max\left(0,\, d^S_{Q,S} - \sqrt{\tfrac{1}{2(m-2)}\left[8+\ln\tfrac{\xi(m-2)}{\delta/2}\right]}\right),\\
\underline{\mu_1} &\stackrel{\text{def}}{=} \max\left(0,\, \mu_1(M^S_{Q,S}) - \sqrt{\tfrac{2}{m-1}\left[4+\ln\tfrac{\xi(m-1)}{\delta/2}\right]}\right),\\
\overline{\mu_2} &\stackrel{\text{def}}{=} \min\left(1,\, \mu_2(M^S_{Q,S}) + \sqrt{\tfrac{2}{m-2}\left[8+\ln\tfrac{\xi(m-2)}{\delta/2}\right]}\right).
\end{aligned}$$

Proof The proof is almost identical to the one of PAC-Bound 3, except that it relies on sample-compressed PAC-Bayesian bounds. Indeed, the inequality is a consequence of Theorem 11, as well as Theorems 41 and 42. The equality $1-\frac{(1-2\cdot\overline{r})^2}{1-2\cdot\underline{d}} = 1-\frac{(\underline{\mu_1})^2}{\overline{\mu_2}}$ is a direct application of Equation (42). ■

PAC-Bounds 3 and 3' are expressed in two forms. The first form relies on bounds on the Gibbs risk and the expected disagreement (denoted $\overline{r}$ and $\underline{d}$). The second form relies on bounds on the first and second moments of the margin (denoted $\underline{\mu_1}$ and $\overline{\mu_2}$). This latter form is used to justify the learning algorithm presented in Section 8.

8. MinCq: Learning by Minimizing the $\mathcal{C}$-bound

In this section, we propose a new algorithm, that we call MinCq, for constructing a weighted majority vote of voters. One version of this algorithm is designed for the supervised inductive framework and minimizes the $\mathcal{C}$-bound. A second version of MinCq that minimizes the $\mathcal{C}$-bound in the transductive (or semi-supervised) setting can be found in Laviolette et al. (2011). Both versions can be expressed as quadratic programs on positive semi-definite matrices.

As is the case for Boosting algorithms (Schapire and Singer, 1999), MinCq is designed to output a Q-weighted majority vote of voters that perform rather poorly individually and, consequently, are often called weak learners. Hence, the decision of each vote is based on a small majority (*i.e.*, with a Gibbs risk just a bit lower than 1/2). Recall that in situations where the Gibbs risk is high (*i.e.*, the first moment of the margin is close to 0), the $\mathcal{C}$-bound can nevertheless remain small if the voters of the majority vote are maximally uncorrelated.

Unfortunately, minimizing the empirical value of the $\mathcal{C}$-bound tends to overfit the data. To overcome this problem, MinCq uses a distribution Q of voters which is constrained to be quasi-uniform (see Equation 37) and for which the first moment of the margin is forced

to be not too close to 0. More precisely, the value $\mu_1(M_Q^S)$ is constrained to be bigger than some strictly positive constant μ. This μ then becomes a hyperparameter of the algorithm that has to be fixed by cross-validation, as the parameter C is for SVM. This new learning strategy is justified by PAC-Bound 3, dedicated to quasi-uniform posteriors[16], and PAC-Bound 3', that is specialized to kernel voters. Hence, MinCq can be viewed as the algorithm that simply looks for the majority vote of margin at least μ that minimizes PAC-Bound 3 (or PAC-Bound 3' in the sample compression case).

MinCq is also justified by two important properties of quasi-uniform majority votes. First, as we shall see in Theorem 43, there is no generality loss when restricting ourselves to quasi-uniform distributions. Second, as we shall see in Theorem 44, for any margin threshold $\mu > 0$ and any quasi-uniform distribution Q such that $\mu_1(M_Q^S) \geq \mu$, there is another quasi-uniform distribution Q' whose margin is exactly μ that achieves the same majority vote and therefore has the same $\mathcal{C}$-bound value.

Thus, to minimize the $\mathcal{C}$-bound, the learner must substantially reduce the variance of the margin distribution – *i.e.*, $\mu_2(M_Q^S)$ – while maintaining its first moment – *i.e.*, $\mu_1(M_Q^S)$ – over the threshold μ. Many learning algorithms actually exploit this strategy in different ways. Indeed, the variance of the margin distribution is controlled by Breiman (2001) for producing random forests, by Dredze et al. (2010) in the transfer learning setting, and by Shen and Li (2010) in the Boosting setting. Thus, the idea of minimizing the variance of the margin is well-known and used. We propose a new theoretical justification for all these types of algorithms and propose a novel learning algorithm, called MinCq, that directly minimizes the $\mathcal{C}$-bound.

8.1 From the $\mathcal{C}$-bound to the MinCq Learning Algorithm

We only consider learning algorithms that construct majority votes based on a (finite) self-complemented hypothesis space $\mathcal{H} = \{f_1, \ldots, f_{2n}\}$ of real-valued voters. Recall that these voters can be classifiers such as decision stumps or can be given by a kernel k evaluated on the examples of S such as $f_i(\cdot) = k(x_i, \cdot)$.

We consider the second form of the $\mathcal{C}$-bound, which relies on the first two moments of the margin of the majority vote classifier (see Theorem 11):

$$\mathcal{C}_Q^{D'} \;=\; 1 - \frac{\left(\mu_1(M_Q^{D'})\right)^2}{\mu_2(M_Q^{D'})}\,.$$

Our first attempts to minimize the $\mathcal{C}$-bound confronted us with two problems.

Problem 1: an empirical $\mathcal{C}$-bound minimization without any regularization tends to overfit the data.

Problem 2: most of the time, the distributions Q minimizing the $\mathcal{C}$-bound $\mathcal{C}_Q^S$ are such that both $\mu_1(M_Q^S)$ and $\mu_2(M_Q^S)$ are very close to 0. Since $\mathcal{C}_Q^S = 1 - (\mu_1(M_Q^S))^2/\mu_2(M_Q^S)$, this gives a 0/0 numerical instability. Since $(\mu_1(M_Q^D))^2/\mu_2(M_Q^D)$ can only be empirically estimated by $(\mu_1(M_Q^S))^2/\mu_2(M_Q^S)$, Problem 2 amplifies Problem 1.

16. PAC-Bound 3 is dedicated to posteriors Q that are aligned on a prior distribution P, but in this section we always consider that the prior distribution P is uniform, thus leading to a quasi-uniform posterior Q.

A natural way to resolve Problem 1 is to restrict ourselves to quasi-uniform distributions, *i.e.*, distributions that are aligned on the uniform prior (see Section 6.1 for the definition). In Section 6, we show that with such distributions, one can upper-bound the Bayes risk without needing a KL-regularization term. Hence, according to this PAC-Bayesian theory, these distributions have some "built-in" regularization effect that should prevent overfitting. Section 7 generalizes these results to the sample compression setting, which is necessary in the case where voters such as kernels are defined using the training set.

The next theorem shows that this restriction on Q does not reduce the set of possible majority votes.

Theorem 43 *Let $\mathcal{H}$ be a self-complemented set. For all distributions Q on $\mathcal{H}$, there exists a quasi-uniform distribution Q' on $\mathcal{H}$ that gives the same majority vote as Q, and that has the same empirical and true $\mathcal{C}$-bound values, i.e.,*

$$B_{Q'} = B_Q\,, \quad \mathcal{C}^S_{Q'} = \mathcal{C}^S_Q \quad \textit{and} \quad \mathcal{C}^D_{Q'} = \mathcal{C}^D_Q\,.$$

Proof Let Q be a distribution on $\mathcal{H}=\{f_1,\ldots,f_{2n}\}$, let $M \stackrel{\text{def}}{=} \max_{i\in\{1,..,n\}}|Q(f_{i+n}) - Q(f_i)|$, and let Q' be defined as

$$Q'(f_i) \stackrel{\text{def}}{=} \frac{1}{2n} + \frac{Q(f_i) - Q(f_{i+n})}{2nM}\,,$$

where the indices of f are defined modulo $2n$ (*i.e.*, $f_{(i+n)+n} = f_i$). Then it is easy to show that Q' is a quasi-uniform distribution. Moreover, for any example $x \in \mathcal{X}$, we have

$$\begin{aligned}
\mathop{\mathbf{E}}_{f\sim Q'} f(x) &\stackrel{\text{def}}{=} \sum_{i=1}^{2n} Q'(f_i)\, f_i(x) \;=\; \sum_{i=1}^{n} (Q'(f_i) - Q'(f_{i+n}))\, f_i(x) \\
&= \sum_{i=1}^{n} \frac{2Q(f_i) - 2Q(f_{i+n})}{2nM}\, f_i(x) \;=\; \frac{1}{nM}\sum_{i=1}^{2n} Q(f_i)\, f_i(x) \\
&= \frac{1}{nM}\mathop{\mathbf{E}}_{f\sim Q} f(x)\,.
\end{aligned}$$

Since $nM > 0$, this implies that $B_{Q'}(x) = B_Q(x)$ for all $x \in \mathcal{X}$. It also shows that $M_{Q'}(x,y) = \frac{1}{nM} M_Q(x,y)$, which implies that $\left(\mu_1(M^{D'}_{Q'})\right)^2 = \left(\frac{1}{nM}\mu_1(M^{D'}_Q)\right)^2$ and $\mu_2(M^{D'}_{Q'}) = \left(\frac{1}{nM}\right)^2\mu_2(M^{D'}_Q)$ for both $D' = D$ and $D' = S$.

The theorem then follows from the definition of the $\mathcal{C}$-bound. ∎

Theorem 43 points out a nice property of the $\mathcal{C}$-bound: different distributions Q that give rise to a same majority vote have the same (real and empirical) $\mathcal{C}$-bound values. Since the $\mathcal{C}$-bound is a bound on majority votes, this is a suitable property. Moreover, PAC-Bounds 3 and 3', together with Theorem 43, indicate that restricting ourselves to quasi-uniform distributions is a natural solution to the problem of overfitting (see Problem 1). Unfortunately, Problem 2 remains since a consequence of the next theorem is that, among all the posteriors Q that minimize the $\mathcal{C}$-bound, there is always one whose empirical margin $\mu_1(M^S_Q)$ is as close to 0 as we want.

Theorem 44 *Let $\mathcal{H}$ be a self-complemented set. For all $\mu \in (0,1]$ and for all quasi-uniform distributions Q on $\mathcal{H}$ having an empirical margin $\mu_1(M_Q^S) \geq \mu$, there exists a quasi-uniform distribution Q' on $\mathcal{H}$, having an empirical margin equal to μ, such that Q and Q' induce the same majority vote and have the same empirical and true $\mathcal{C}$-bound values, i.e.,*

$$\mu_1(M_{Q'}^S) = \mu\,, \quad B_{Q'} = B_Q\,, \quad \mathcal{C}_{Q'}^S = \mathcal{C}_Q^S \quad \textit{and} \quad \mathcal{C}_{Q'}^D = \mathcal{C}_Q^D.$$

Proof Let Q be a quasi-uniform distribution on $\mathcal{H}=\{f_1,\ldots,f_{2n}\}$ such that $\mu_1(M_Q^S) \geq \mu$. We define Q' as

$$Q'(f_i) \stackrel{\text{def}}{=} \frac{\mu}{\mu_1(M_Q^S)} \cdot Q(f_i) + \left(1 - \frac{\mu}{\mu_1(M_Q^S)}\right) \cdot 1/2n\,, \quad i \in \{1,..,2n\}\,.$$

Clearly Q' is a quasi-uniform distribution since it is a convex combination of a quasi-uniform distribution and the uniform one. Then, similarly as in the proof of Theorem 43, one can easily show that $\mathop{\mathbf{E}}_{f\sim Q'} f(x) = \frac{\mu}{\mu_1(M_Q^S)} \mathop{\mathbf{E}}_{f\sim Q} f(x)$, which implies the result. ■

Training set bounds (such as VC-bounds for example) are known to degrade when the capacity of classification increases. As shown by Theorem 44 for the majority vote setting, this capacity increases as μ decreases to 0. Thus, we expect that any training set bound degrades for small μ. This is not the case for the $\mathcal{C}$-bound itself, but the $\mathcal{C}$-bound is not a training set bound. To obtain a training set bound, we have to relate the empirical value $\mathcal{C}_Q^S$ to the true one $\mathcal{C}_Q^D$, which is done via PAC-Bounds 3 and 3'. In these bounds, there is indeed a degradation as μ decreases because the true $\mathcal{C}$-bound is of the form $1-(\mu_1(M_Q^D))^2/\mu_2(M_Q^D)$. Since $\mu = \mu_1(M_Q^S)$, and because a small $\mu_1(M_Q^S)$ tends to produce small $\mu_2(M_Q^S)$, the bounds on $\mathcal{C}_Q^D$ given $\mathcal{C}_Q^S$ that outcomes from PAC-Bounds 3 and 3' are therefore much looser for small μ because of the 0/0 instability. As explained in the introduction of the present section, one way to overcome the instability identified in Problem 2 is to restrict ourselves to quasi-uniform distributions whose empirical margin is greater or equal than some threshold μ. Interestingly, thanks to Theorem 44, this is equivalent to restricting ourselves to distributions having empirical margin *exactly equal to* μ. From Theorems 11 and 44, it then follows that *minimizing the $\mathcal{C}$-bound, under the constraint $\mu_1(M_Q^S) \geq \mu$, is equivalent to minimizing $\mu_2(M_Q^S)$, under the constraint $\mu_1(M_Q^S) = \mu$*, from this observation, and the fact that minimizing PAC-Bounds 3 and 3' is equivalent to minimizing the empirical $\mathcal{C}$-bound $\mathcal{C}_Q^S$, we can now define the algorithm MinCq.

In this section, μ always represents a restriction on the margin. Moreover, we say that a value μ is *D'-realizable* if there exists some quasi-uniform distribution Q such that $\mu_1(M_Q^{D'}) = \mu$. The proposed algorithm, called MinCq, is then defined as follows.

Definition 45 (MinCq Algorithm) Given a self-complemented set $\mathcal{H}$ of voters, a training set S, and a S-realizable $\mu > 0$, among all quasi-uniform distributions Q of empirical margin $\mu_1(M_Q^S)$ exactly equal to μ, the algorithm MinCq consists in finding one that minimizes $\mu_2(M_Q^S)$.

This algorithm can be translated as a simple quadratic program (QP) that has only n variables (instead of $2n$), and thus can be easily solved by any QP solver. In the next subsection, we explain how the algorithm of Definition 45 can be turned into a QP.

8.2 MinCq as a Quadratic Program

Given a training set S, and a self-complemented set $\mathcal{H}$ of voters $\{f_1, f_2, \ldots, f_{2n}\}$, let

$$\mathcal{M}_i \stackrel{\text{def}}{=} \mathop{\mathbf{E}}_{(x,y)\sim S} y\, f_i(x) \qquad \text{and} \qquad \mathcal{M}_{i,j} \stackrel{\text{def}}{=} \mathop{\mathbf{E}}_{(x,y)\sim S} f_i(x)\, f_j(x)\,.$$

Let $\mathbf{M}$ be a symmetric $n \times n$ matrix, $\mathbf{a}$ be a column vector of n elements, and $\mathbf{m}$ be a column vector of n elements defined by

$$\mathbf{M} \stackrel{\text{def}}{=} \begin{bmatrix} \mathcal{M}_{1,1} & \mathcal{M}_{1,2} & \ldots & \mathcal{M}_{1,n} \\ \mathcal{M}_{2,1} & \mathcal{M}_{2,2} & \ldots & \mathcal{M}_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{M}_{n,1} & \mathcal{M}_{n,2} & \ldots & \mathcal{M}_{n,n} \end{bmatrix}, \quad \mathbf{a} \stackrel{\text{def}}{=} \begin{bmatrix} \frac{1}{n}\sum_{i=1}^n \mathcal{M}_{i,1} \\ \frac{1}{n}\sum_{i=1}^n \mathcal{M}_{i,2} \\ \vdots \\ \frac{1}{n}\sum_{i=1}^n \mathcal{M}_{i,n} \end{bmatrix}, \quad \text{and} \quad \mathbf{m} \stackrel{\text{def}}{=} \begin{bmatrix} \mathcal{M}_1 \\ \mathcal{M}_2 \\ \vdots \\ \mathcal{M}_n \end{bmatrix}. \tag{43}$$

Finally, let $\mathbf{q}$ be the column vector of n QP-variables, where each element q_i represents the weight $Q(f_i)$.

Using the above definitions and the fact that $\mathcal{H}$ is self-complemented, one can show that

$$\mathcal{M}_{i+n} = -\mathcal{M}_i\,, \quad \mathcal{M}_{i+n,j} = \mathcal{M}_{i,j+n} = -\mathcal{M}_{i,j}\,, \quad \text{and} \quad q_{i+n} = \frac{1}{n} - q_i\,.$$

Moreover, it follows from the definitions of the first two moments of the margin $\mu_1(M_Q^S)$ and $\mu_2(M_Q^S)$ (see Equations 6 and 8) that

$$\mu_1(M_Q^S) \;=\; \sum_{i=1}^{2n} q_i\, \mathcal{M}_i\,, \quad \text{and} \quad \mu_2(M_Q^S) \;=\; \sum_{i=1}^{2n}\sum_{j=1}^{2n} q_i q_j\, \mathcal{M}_{i,j}\,.$$

As MinCq consists in finding the quasi-uniform distribution Q that minimizes $\mu_2(M_Q^S)$, with a margin $\mu_1(M_Q^S)$ exactly equal to the hyperparameter μ, let us now rewrite $\mu_2(M_Q^S)$ and $\mu_1(M_Q^S)$ using the vectors and matrices defined in Equation (43). It follows that

$$\begin{aligned}
\mu_2(M_Q^S) \;=\; \sum_{i=1}^{2n}\sum_{j=1}^{2n} q_i q_j\, \mathcal{M}_{i,j} \;&=\; \sum_{i=1}^{n}\sum_{j=1}^{n} \Big[q_i q_j - q_{i+n} q_j - q_i q_{j+n} + q_{i+n} q_{j+n}\Big] \mathcal{M}_{i,j} \\
&=\; \sum_{i=1}^{n}\sum_{j=1}^{n} \left[4 q_i q_j - \frac{4}{n}\, q_i + \frac{1}{n^2}\right] \mathcal{M}_{i,j} \\
&=\; 4 \sum_{i=1}^{n}\sum_{j=1}^{n} q_i q_j\; \mathcal{M}_{i,j} - \frac{4}{n} \sum_{i=1}^{n}\sum_{j=1}^{n} q_i\; \mathcal{M}_{i,j} + \frac{1}{n^2} \sum_{i=1}^{n}\sum_{j=1}^{n} \mathcal{M}_{i,j} \\
&=\; 4\Big(\mathbf{q}^\top \mathbf{M}\, \mathbf{q} \;-\; \mathbf{a}^\top\, \mathbf{q}\Big) \;+\; \frac{1}{n^2} \sum_{i=1}^{n}\sum_{j=1}^{n} \mathcal{M}_{i,j}\,, \qquad (44)
\end{aligned}$$

and

$$\mu_1(M_Q^S) \;=\; \sum_{i=1}^{2n} q_i \mathcal{M}_i \;=\; \sum_{i=1}^{n} (q_i - q_{i+n}) \mathcal{M}_i \;=\; \sum_{i=1}^{n} \left(2q_i - \frac{1}{n}\right) \mathcal{M}_i \;=\; 2\sum_{i=1}^{n} q_i \, \mathcal{M}_i - \frac{1}{n}\sum_{i=1}^{n} \mathcal{M}_i$$
$$= \; 2\mathbf{m}^\top \mathbf{q} - \frac{1}{n}\sum_{i=1}^{n} \mathcal{M}_i \,.$$

As the objective function $\mu_2(M_Q^S)$ and the constraint $\mu_1(M_Q^S) = \mu$ of the QP can be defined using only n variables, there is no need to consider in the QP the weights of the last n voter. These weights can always be recovered from the n first, because $q_{i+n} = \frac{1}{n} - q_i$, for any i. Note however that to be sure that the solution of the QP has the quasi-uniformity property, we have to add the following constraints to the program:

$$q_i \;\in\; [0, \tfrac{1}{n}] \qquad\qquad \text{for any } \; i \,.$$

Note that the multiplicative constant 4 and the additive constant $\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\mathcal{M}_{i,j}$ from Equation (44) can be omitted, as the optimal solution will stay the same. From all that precedes and given any S-realizable μ, MinCq solves the optimization problem described by Program 1.

Program 1 : **MinCq** - *a quadratic program for classification*

Solve $\quad \text{argmin}_{\mathbf{q}} \quad \mathbf{q}^\top \, \mathbf{M} \, \mathbf{q} \; - \; \mathbf{a}^\top \, \mathbf{q}$

under constraints : $\mathbf{m}^\top \, \mathbf{q} = \frac{\mu}{2} + \frac{1}{2n}\sum_{i=1}^{n} \mathcal{M}_i$

and : $0 \le q_i \le \frac{1}{n} \quad \forall i \in \{1, \ldots, n\}$

To prove that Program 1 is a quadratic program, it suffices to show that $\mathbf{M}$ is a positive semi-definite matrix. This is a direct consequence of the fact that each $\mathcal{M}_{i,j}$ can be viewed as a scalar product, since

$$\mathcal{M}_{i,j} = \left(\sqrt{\tfrac{1}{|S|}}\, f_i(x)\right)_{x \in S_{\mathcal{X}}} \;\cdot\; \left(\sqrt{\tfrac{1}{|S|}}\, f_j(x)\right)_{x \in S_{\mathcal{X}}}, \quad \text{where } S_{\mathcal{X}} \stackrel{\text{def}}{=} \{x \colon (x, y) \in S\}.$$

Finally, the Q-weighted majority vote output by MinCq is

$$B_Q(x) \;=\; \text{sgn}\left[\mathop{\mathbf{E}}_{f \sim Q} f(x)\right] \;=\; \text{sgn}\left[\sum_{i=1}^{2n} q_i f_i(x)\right] \;=\; \text{sgn}\left[\sum_{i=1}^{n} q_i f_i(x) + \sum_{i=n+1}^{2n} q_i f_i(x)\right]$$
$$= \; \text{sgn}\left[\sum_{i=1}^{n} q_i f_i(x) + \sum_{i=1}^{n} (\tfrac{1}{n} - q_i) \cdot -f_i(x)\right]$$
$$= \; \text{sgn}\left[\sum_{i=1}^{n} (2q_i - \tfrac{1}{n}) f_i(x)\right].$$

8.3 Experiments

We now compare MinCq to state-of-the-art learning algorithms in three different contexts: *handwritten digits recognition*, *classical binary classification tasks*, and *Amazon reviews sentiment analysis*. A *context* (Lacoste et al., 2012) represents a distribution on the different tasks a learning algorithm can encounter, and a sample from a context is a collection of data sets.

For each context, each data set is randomly split into a training set S and a testing set T. When hyperparameters have to be chosen for an algorithm, 5-fold cross-validation is run on the training set S, and the hyperparameter values that minimize the mean cross-validation risk are chosen. Using these values, the algorithm is trained on the whole training set S, and then evaluated on the testing set T.

For the first two contexts, we compare MinCq using decision stumps as voters (referred to as StumpsMinCq), MinCq using RBF kernel functions $k(x, x') = \exp(-\gamma||x - x'||^2)$ as voters (referred to as RbfMinCq), AdaBoost (Schapire and Singer, 1999) using decision stumps (referred to as StumpsAdaBoost), and the soft-margin Support Vector Machine (SVM) (Cortes and Vapnik, 1995) using the RBF kernel, referred to as RbfSVM. For the last context, we compare MinCq using linear kernel functions $k(x, x') = x \cdot x'$ as voters (referred to as LinearMinCq), and the SVM using the same linear kernel, referred to as LinearSVM.

For the three variants of MinCq, the quadratic program is solved using CVXOPT (Dahl and Vandenberghe, 2007), an off-the-shelf convex optimization solver.

StumpsAdaBoost: For StumpsAdaBoost, we use decision stumps as weak learners. For each attribute, 10 decision stumps (and their complement) are generated, for a total of 20 decision stumps per attribute. The number of boosting rounds is chosen among the following 15 values: 10, 25, 50, 75, 100, 125, 150, 175, 200, 225, 250, 275, 500, 750 and 1000.

StumpsMinCq: For StumpsMinCq, we use the same 10 decision stumps per attribute as for StumpsAdaBoost. Note that we do not need to consider the complement stumps in this case, as MinCq automatically considers self-complemented sets of voters. MinCq's hyperparameter μ is chosen among 15 values between 10^{-4} and 10^{0} on a logarithmic scale.

RbfSVM: The γ hyperparameter of the RBF kernel and the C hyperparameter of the SVM are chosen among 15 values between 10^{-4} and 10^{1} for γ, and among 15 values between 10^{0} and 10^{8} for C, both on a logarithmic scale.

RbfMinCq: For RbfMinCq, we consider 15 values of μ between 10^{-4} and 10^{-2} on a logarithmic scale, and the same 15 values of γ as in SVM for the RBF kernel voters.

LinearSVM: When using the linear kernel, the C parameter of the SVM is chosen among 15 values between 10^{-4} and 10^{2}, on a logarithmic scale. All SVM experiments are done using the implementation of Pedregosa et al. (2011).

LinearMinCq: For LinearMinCq, we consider 15 values of μ between 10^{-4} and 10^{-2} on a logarithmic scale.

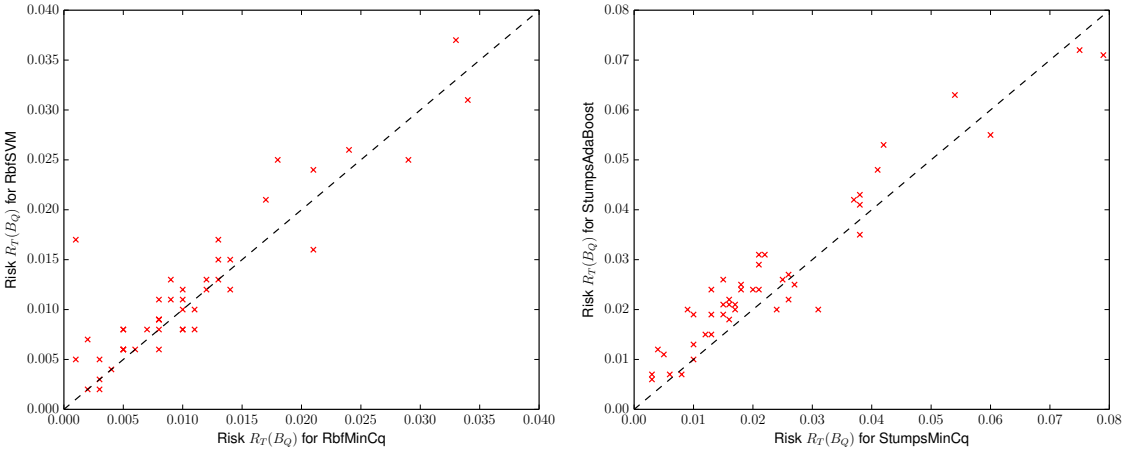


Figure 8: Comparison of the risks on the testing set for each algorithm and each MNIST binary data set. The figure on the left shows a comparison of the risks of RbfMinCq (x-axis) and RbfSVM (y-axis). The figure on the right compares StumpsMinCq (x-axis) and StumpsAdaBoost (y-axis). On each scatter plot, a point represents a pair of risks for a particular MNIST binary data set. A point above the diagonal line indicates better performance for MinCq.

Statistical Comparison Tests		
	RbfMinCq vs RbfSVM	StumpsMinCq vs StumpsAdaBoost
Poisson binomial test	88%	99%
Sign test (p-value)	0.01	0.00

Table 2: Statistical tests comparing MinCq to either RbfSVM or StumpsAdaBoost. The Poisson binomial test gives the probability that MinCq has a better performance than another algorithm on this context. The sign test gives a p-value representing the probability that the null hypothesis is true (*i.e.*, MinCq and the other algorithm both have the same performance on this context).

When using the RBF kernel for the SVM or MinCq, each data set is normalized using a hyperbolic tangent. For each example x, each attribute $x_1, x_2, \ldots, x_n$ is renormalized with $x_i' = \tanh\left[\frac{x_i - \overline{x_i}}{\sigma_i}\right]$, where $\overline{x_i}$ and σ_i are the mean and standard deviation of the i^{th} attribute respectively, calculated on the training set S. Normalizing the features when using the RBF kernel is a common practice and gives better results for both MinCq and SVM. Empirically, we observe that the performance gain of RbfMinCq with normalized data is even more significant than for RbfSVM.

8.3.1 Handwritten Digits Recognition Context

The first context of interest to compare MinCq with other learning algorithms is the handwritten digits recognition. For this task, we use the *MNIST database of handwritten digits* of Lecun and Cortes. We split the original data set into 45 binary classification tasks, where the union of all binary data sets recovers the original data set, and the intersection of any pair of binary data sets gives the empty set. Therefore, any example from the original data set appears on one and only one binary data set, thus avoiding any correlation between the binary data sets. For each resulting binary data set, we randomly choose 500 examples to be in the training set S, and the testing set T consists of the remaining examples. Figure 8 shows the resulting test risk for each binary data set and each algorithm.

Table 2 shows two statistical tests to compare the algorithms on the handwritten digits recognition context: the Poisson binomial test (Lacoste et al., 2012) and the sign test (Mendenhall, 1983). Both methods suggest that RbfMinCq outperforms RbfSVM on this context, and that StumpsMinCq outperforms StumpsAdaBoost.

8.3.2 Classical Binary Classification Tasks Context

This second context of interest is a more general one: it consists of multiple binary classification data sets coming from the UCI Machine Learning Repository (Blake and Merz, 1998). These data sets are commonly used as a benchmark for learning algorithms, and may help to answer the question "How well may a learning algorithm perform on many unrelated classification tasks". For each data set, half of the examples (up to a maximum of 500) are randomly chosen to be in the training set S, and the remaining examples are in the testing set T. Table 3 shows the resulting test risks on this context, for each algorithm.

Table 3 also shows a statistical comparison of all algorithms on the classical binary classification tasks context, using the Poisson binomial test and the sign test. On this context, both statistical tests show no significant performance difference between RbfMinCq and RbfSVM, and between StumpsMinCq and StumpsAdaBoost, implying that these pairs of algorithms perform similarly well on this general context.

8.3.3 Amazon Reviews Sentiment Analysis

This context contains 4 sentiment analysis data sets, representing product types (*books*, *DVDs*, *electronics* and *kitchen appliances*). The task is to learn from an Amazon.com product user review in natural language, and predict the *polarity* of the review, that is either negative (3 stars or less) or positive (4 or 5 stars). The data sets come from Blitzer et al. (2007), where the natural language reviews have already been converted into a set of *unigrams* and *bigrams* of terms, with a count. For each data set, a training set of 1000 positive reviews and 1000 negative reviews are provided, and the remaining reviews are available in a testing set. The original feature space of these data sets is between $90,000$ and $200,000$ dimensions. However, as most of the unigrams and bigrams are not significant and to reduce the dimensionality, we only consider unigrams and bigrams that appear at least 10 times on the training set (as in Chen et al., 2011), reducing the numbers of dimensions to between 3500 and 6000. Again as in Chen et al. (2011), we apply standard *tf-idf* feature re-weighting (Salton and Buckley, 1988). Table 4 shows the resulting test risks for each algorithm.

Data Set Information			Risk $R_T(B_Q)$ for Each Algorithm			
Name	$\|S\|$	$\|T\|$	RbfMinCq	RbfSVM	StumpsMinCq	StumpsAdaBoost
Australian	345	345	0.142	**0.133**	**0.165**	0.168
Balance	313	312	0.054	**0.042**	0.042	**0.032**
BreastCancer	350	349	**0.037**	0.046	**0.037**	0.060
Car	500	1228	0.074	**0.032**	0.320	**0.291**
Cmc	500	973	**0.303**	0.306	0.140	**0.134**
Credit-A	345	345	**0.122**	0.133	**0.304**	0.308
Cylinder	270	270	**0.204**	0.233	**0.125**	0.148
Ecoli	168	168	0.077	**0.071**	**0.289**	**0.289**
Flags	97	97	**0.289**	0.320	**0.071**	**0.071**
Glass	107	107	**0.206**	**0.206**	**0.268**	0.309
Heart	135	135	0.163	**0.156**	**0.262**	0.271
Hepatitis	78	77	0.169	**0.143**	**0.185**	**0.185**
Horse	184	184	**0.185**	0.196	**0.169**	0.221
Ionosphere	176	175	0.114	**0.069**	0.245	**0.174**
Letter:AB	500	1055	0.007	**0.003**	**0.109**	0.120
Letter:DO	500	1058	0.021	**0.018**	**0.005**	0.010
Letter:OQ	500	1036	**0.023**	0.036	**0.020**	0.048
Liver	173	172	**0.267**	0.285	**0.042**	0.052
Monks	216	216	0.245	**0.208**	0.306	**0.236**
Nursery	500	12459	**0.025**	0.026	**0.025**	0.026
Optdigits	500	3323	0.034	**0.027**	**0.089**	**0.089**
Pageblock	500	4973	**0.045**	0.048	0.059	**0.055**
Pendigits	500	6994	**0.007**	0.008	**0.069**	0.084
Pima	384	384	**0.253**	0.255	0.273	**0.250**
Segment	500	1810	**0.017**	0.018	0.040	**0.022**
Spambase	500	4101	**0.067**	0.077	0.133	**0.070**
Tic-tac-toe	479	479	0.033	**0.025**	**0.330**	0.353
USvote	218	217	**0.051**	**0.051**	**0.051**	**0.051**
Wine	89	89	**0.034**	0.045	0.169	**0.034**
Yeast	500	984	0.286	**0.279**	0.324	**0.306**
Zoo	51	50	**0.040**	0.060	0.060	**0.040**

Statistical Comparison Tests		
	RbfMinCq vs RbfSVM	StumpsMinCq vs StumpsAdaBoost
Poisson binomial test	54%	48%
Sign test (p-value)	0.36	0.35

Table 3: Risk on the testing set for all algorithms, on the classical binary classification task context. See Table 2 for an explanation of the statistical tests.

Table 4 also shows a statistical comparison of the algorithms on this context, again using the Poisson binomial test and the sign test. LinearMinCq has an edge over LinearSVM, as it wins or draws on each data set. However, both statistical tests show no significant performance difference between LinearMinCq and LinearSVM.

These experiments show that minimizing the $\mathcal{C}$-bound, and thus favoring majority votes for which the voters are maximally uncorrelated, is a sound approach. MinCq is very competitive with both AdaBoost and the SVM on the classical binary tasks context and the Amazon reviews sentiment analysis context. MinCq even shows a highly significant performance gain on the handwritten digits recognition context, implying that on certain types of tasks or data sets, minimizing the $\mathcal{C}$-bound offers a state-of-the-art performance.

Data Set Information			Risk $R_T(B_Q)$ for Each Algorithm	
Name	$\lvert S\rvert$	$\lvert T\rvert$	LinearMinCq	LinearSVM
Books	2000	4465	**0.158**	**0.158**
DVD	2000	3586	**0.162**	0.163
Kitchen	2000	5945	**0.130**	0.131
Electronics	2000	5681	**0.116**	0.118

Statistical Comparison Tests	
	LinearMinCq vs LinearSVM
Poisson binomial test	68%
Sign test (p-value)	0.31

Table 4: Risk on the testing set for all algorithms, on the Amazon reviews sentiment analysis context. See Table 2 for an explanation of the statistical tests.

However, for all above experiments, we observe that the empirical values of the PAC-Bounds are trivial (close to 1). Remember that, inspired by PAC-Bounds 3 and 3', the MinCq algorithm learns the weights of a majority vote by minimizing the second moment of the margin while fixing its first moment μ to some value. In these experiments, the value of μ chosen by cross-validation is always very close to 0 (basically, $\mu = 10^{-4}$). This implies that $\mathcal{C}_Q^S = 1 - \frac{\mu^2}{\mu_2(M_Q^S)}$ is very close to the $1 - \frac{0}{0}$ form, leading to a severe degradation of PAC-Bayesian bounds for $\mathcal{C}_Q^D$. Note that the voters were all *weak* in the former experiments. This explains why very small values of μ were selected by cross-validation.

8.3.4 Experiments with Stronger Voters

In the following experiment, we show that one can obtain much better bound values by using *stronger* voters, that is, voters with a better individual performance. To do so, instead of considering decision stumps, we consider decision trees.[17] We use 100 decision tree classifiers generated with the implementation of Pedregosa et al. (2011) (we set the maximum depth to 10 and the number of features per node to 1). By using these strong voters, it is possible to achieve higher values of μ.[18]

Figure 9 shows the empirical $\mathcal{C}$-bound value and its corresponding PAC-Bayesian bound values for multiple values of μ on the Mushroom UCI data set. From the 8124 examples, 500 have been used to construct the set of voters, 4062 for the training set, and the remaining examples for the testing set. The figure shows the PAC-Bayesian bounds get tighter when μ is increasing. Note however that the empirical $\mathcal{C}$-bound slightly increases from 0.001 to 0.016. The risk on the testing set of the majority vote (not shown in the figure) is 0 for most values of μ, but also increases a bit for the highest values (remaining below 0.001).

17. A decision stump can be seen as a (weak) decision tree of depth 1.
18. Note that the set of decision trees was learned on a *fresh* set of examples, disjoint from the training data. We do so to ensure that all computed PAC-Bounds are valid, even if they are not designed to handle *sample-compressed* voters.

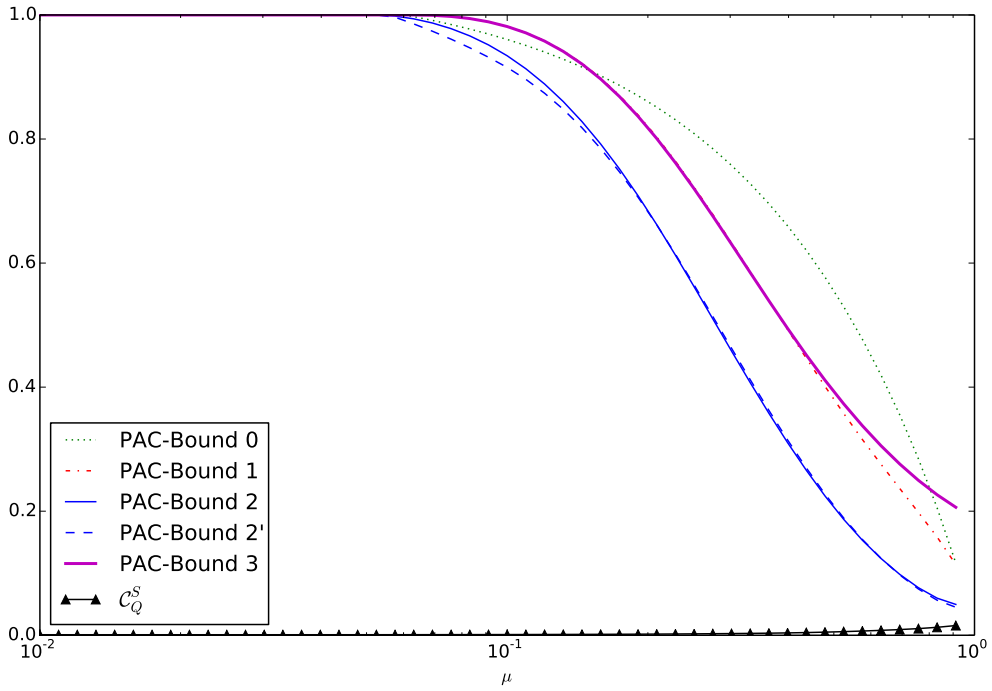


Figure 9: Values of empirical $\mathcal{C}$-bound and corresponding PAC-Bounds 0, 1, 2, 2' and 3 on the majority votes output by MinCq, for multiple values of μ.

Hence, we obtain tight bounds for high values of μ (PAC-Bounds 2 and 2' are under 0.2). Nevertheless, these PAC-Bayesian bounds are not tight enough to precisely guide the selection of μ. This is why we rely on cross-validation to select a good value of μ.

Finally, we also see that PAC-Bound 3 is looser than other bounds over $\mathcal{C}_Q^D$, but this was expected as it was not designed to be as tight as possible. That being said, PAC-Bound 3 has the same behavior than PAC-Bounds 1 and 2. This suggests that we can rely on it to justify the MinCq learning algorithm once the hyperparameter μ is fixed.

9. Conclusion

In this paper, we have revisited the work presented in Lacasse et al. (2006) and Laviolette et al. (2011). We clarified the presentation of previous results and extended them, as well as actualizing the discussion regarding the ever growing development of PAC-Bayesian theory.

We have derived a risk bound (called the $\mathcal{C}$-bound) for the weighted majority vote that depends on the first and the second moment of the associated margin distribution (Theorem 11). The proposed bound is based on the one-sided Chebyshev inequality, which, under the mild condition of Proposition 14, is the tightest inequality for any real-valued random variable given only its first two moments. Also, as shown empirically by Figure 3, this bound has a strong predictive power on the risk of the majority vote.

We have also shown that the original PAC-Bayesian theorem, together with new ones, can be used to obtain high-confidence estimates of this new risk bound that holds uniformly

for *all* posterior distributions. We have generalized these PAC-Bayesian results to the (more general) sample compression setting, allowing one to make use of voters that are constructed with elements of the training data, such as kernel functions $y_i k(x_i, \cdot)$. Moreover, we have presented PAC-Bayesian bounds that have the uncommon property of having no Kullback-Leibler divergence term (PAC-Bounds 3 and 3'). These bounds, together with the $\mathcal{C}$-bound, gave the theoretical foundation to the learning algorithm introduced at the end of the paper, that we have called MinCq. The latter turns out to be expressible in the nice form of a quadratic program. MinCq is not only based on solid theoretical guarantees, it also performs very well on natural data, namely when compared with the state-of-the-art SVM.

This work tackled the simplest problem in machine learning (the supervised binary classification in presence of i.i.d. data), and we now consider that the PAC-Bayesian theory is mature enough to embrace a variety of more sophisticated frameworks. Indeed, in the recent years several authors applied this theory to many more complex paradigms: Transductive Learning (Derbeko et al., 2004; Catoni, 2007; Bégin et al., 2014), Domain Adaptation (Germain et al., 2013), Density Estimation (Seldin and Tishby, 2009; Higgs and Shawe-Taylor, 2010), Structured output Prediction (McAllester, 2007; Giguère et al., 2013; London et al., 2014), Co-clustering (Seldin and Tishby, 2009, 2010), Martingales (Seldin et al., 2012), U-Statistics of higher order (Lever et al., 2013) or other non-i.i.d. settings (Ralaivola et al., 2010), Multi-armed Bandit (Seldin et al., 2011) and Reinforcement Learning (Fard and Pineau, 2010; Fard et al., 2011).

Acknowledgements

This work has been supported by National Science and Engineering Research Council (NSERC) Discovery grants 262067 and 0122405. Computations were performed on the Colosse supercomputer grid at Université Laval, under the auspices of Calcul Québec and Compute Canada. The operations of Colosse are funded by the NSERC, the Canada Foundation for Innovation (CFI), NanoQuébec, and the Fonds de recherche du Québec – Nature et technologies (FRQNT).

We would also like to thank Nicolas Usunier for his insightful comments and participation in the preliminary work on majority votes, and Malik Younsi for solving one of our conjectures leading to the nice $\xi(m)+m$ term in the statement of Theorem 28.

Finally, we sincerely thank the three anonymous reviewers for the exceptional quality of their work.

Appendix A. Auxiliary mathematical results

Lemma 46 (Markov's inequality) *For any random variable X such that $\mathbf{E}(X) = \mu$, and for any $a > 0$, we have*

$$\Pr\left(|X| \geq a\right) \ \leq \ \frac{\mu}{a}\,.$$

Lemma 47 (Jensen's inequality) *For any random variable X and any convex function f, we have*

$$f(\mathbf{E}\,[X]) \ \leq \ \mathbf{E}\,[f(X)]\,.$$

Lemma 48 (One-sided Chebyshev inequality) *For any random variable X such that $\mathbf{E}(X) = \mu$ and $\mathbf{Var}(X) = \sigma^2$, and for any $a > 0$, we have*

$$\Pr\Big(X - \mu \geq a\Big) \;\leq\; \frac{\sigma^2}{\sigma^2 + a^2}\,.$$

Proof First observe that $\Pr\Big(X-\mu \geq a\Big) \leq \Pr\Big(\left[X - \mu + \frac{\sigma^2}{a}\right]^2 \geq \left[a + \frac{\sigma^2}{a}\right]^2\Big)$. Let us now apply Markov's inequality (Lemma 46) to bound this probability. We obtain

$$\begin{aligned}
\Pr\left(\left[X - \mu + \frac{\sigma^2}{a}\right]^2 \geq \left[a + \frac{\sigma^2}{a}\right]^2\right) \;&\leq\; \frac{\mathbf{E}\left[X - \mu + \frac{\sigma^2}{a}\right]^2}{\left[a + \frac{\sigma^2}{a}\right]^2} \qquad \text{(Markov's inequality)} \\
&=\; \frac{\mathbf{E}\,(X-\mu)^2 + 2\left(\frac{\sigma^2}{a}\right)\mathbf{E}\,(X-\mu) + \left(\frac{\sigma^2}{a}\right)^2}{\left[a + \frac{\sigma^2}{a}\right]^2} \\
&=\; \frac{\sigma^2 + \left(\frac{\sigma^2}{a}\right)^2}{\left[a + \frac{\sigma^2}{a}\right]^2} \;=\; \frac{\sigma^2\left(1 + \frac{\sigma^2}{a^2}\right)}{(\sigma^2 + a^2)\left(1 + \frac{\sigma^2}{a^2}\right)} \;=\; \frac{\sigma^2}{\sigma^2 + a^2}\,,
\end{aligned}$$

because $\mathbf{E}\,(X-\mu)^2 = \mathbf{Var}(X) = \sigma^2$ and $\mathbf{E}\,(X-\mu) = \mathbf{E}(X) - \mathbf{E}(X) = 0$. ■

Note that the proof Theorem 49 (below) by Cover and Thomas (1991) considers that probability distributions Q and P are discrete, but their argument is straightforwardly generalizable to continuous distributions.

Theorem 49 (Cover and Thomas, 1991, Theorem 2.7.2) *The Kullback-Leibler divergence* $\mathrm{KL}(Q\|P)$ *is convex in the pair* (Q, P), i.e., *if* (Q_1, P_1) *and* (Q_2, P_2) *are two pairs of probability distributions, then*

$$\mathrm{KL}\big(\,\lambda Q_1 + (1-\lambda)Q_2\,\big\|\,\lambda P_1 + (1-\lambda)P_2\,\big) \;\leq\; \lambda\,\mathrm{KL}\big(Q_1\big\|P_1\big) + (1-\lambda)\,\mathrm{KL}\big(Q_2\big\|P_2\big)\,,$$

for all $\lambda \in [0, 1]$.

Corollary 50 *Both following functions are convex:*

1. *The function* $\mathrm{kl}(q\|p)$ *of Equation* (21), i.e., *the Kullback-Leibler divergence between two Bernoulli distributions;*

2. *The function* $\mathrm{kl}(q_1, q_2\|p_1, p_2)$ *of Equation* (31), i.e., *the Kullback-Leibler divergence between two distributions of trivalent random variables.*

Proof Straightforward consequence of Theorem 49. ■

Lemma 51 (Maurer, 2004) *Let X be any random variable with values in $[0,1]$ and expectation $\mu = \mathbf{E}(X)$. Denote $\mathbf{X}$ the vector containing the results of n independent realizations of X. Then, consider a Bernoulli random variable X' ($\{0,1\}$-valued) of probability of success μ,* i.e., *$\Pr(X'=1) = \mu$. Denote $\mathbf{X}' \in \{0,1\}^n$ the vector containing the results of n independent realizations of X'.*

If function $f : [0,1]^n \to \mathbb{R}$ is convex, then

$$\mathbf{E}\big[f(\mathbf{X})\big] \;\leq\; \mathbf{E}\big[f(\mathbf{X}')\big]\,.$$

The proof of Lemma 52 (below) follows the key steps of the proof of Lemma 51 by Maurer (2004), but we include a few more mathematical details for completeness. Interestingly, the proof highlights that one can generalize Maurer's lemma even more, to embrace random variables of any (countable) number of possible outputs. Note that another generalization of Maurer's lemma is given in Seldin et al. (2012) to embrace the case where the random variables $X_1, \ldots, X_n$ are a martingale sequence instead of being independent.

Lemma 52 (Generalization of Lemma 51) *Let the tuple (X,Y) be a random variable with values in $[0,1]^2$, such that $X+Y \leq 1$, and with expectation $(\mu_X, \mu_Y) = (\mathbf{E}(X), \mathbf{E}(Y))$. Given n independent realizations of (X,Y), denote $\mathbf{X} = (X_1, \ldots, X_n)$ the vector of corresponding X-values and $\mathbf{Y} = (Y_1, \ldots, Y_n)$ the vector of corresponding Y-values. Then, consider a random variable (X',Y') with three possible outcomes, $(1,0)$, $(0,1)$ and $(0,0)$, of expectations μ_X, μ_Y and $1-\mu_X-\mu_Y$, respectively. Denote $\mathbf{X}', \mathbf{Y}' \in \{0,1\}^n$ the vectors of n independent realizations of (X',Y').*

If a function $f : [0,1]^n \times [0,1]^n \to \mathbb{R}$ is convex, then

$$\mathbf{E}\big[f(\mathbf{X},\mathbf{Y})\big] \;\leq\; \mathbf{E}\big[f(\mathbf{X}',\mathbf{Y}')\big]\,.$$

Proof Given two vectors $\mathbf{x} = (x_1, \ldots, x_n), \mathbf{y} = (y_1, \ldots, y_n) \in [0,1]^n$, let us define

$$(\mathbf{x},\mathbf{y}) \;\stackrel{\text{def}}{=}\; \big((x_1,y_1),(x_2,y_2),\ldots,(x_n,y_n)\big) \;\in\; ([0,1]\times[0,1])^n\,.$$

Consider $H = \{(1,0),(0,1),(0,0)\}$. Lemma 53 (below) shows that any point $(\mathbf{x},\mathbf{y})$ can be written as a convex combination of the extreme points $\boldsymbol{\eta} = (\eta_1, \eta_2, \ldots, \eta_n) \in H^n$:

$$(\mathbf{x},\mathbf{y}) \;=\; \sum_{\boldsymbol{\eta}\in H^n} \left[\left(\prod_{i:\eta_i=(1,0)} x_i \right) \left(\prod_{i:\eta_i=(0,1)} y_i \right) \left(\prod_{i:\eta_i=(0,0)} 1-x_i-y_i \right) \right] \cdot \boldsymbol{\eta}\,. \tag{45}$$

Convexity of function f implies

$$f(\mathbf{x},\mathbf{y}) \;\leq\; \sum_{\boldsymbol{\eta}\in H^n} \left[\left(\prod_{i:\eta_i=(1,0)} x_i \right) \left(\prod_{i:\eta_i=(0,1)} y_i \right) \left(\prod_{i:\eta_i=(0,0)} 1-x_i-y_i \right) \right] \cdot f(\boldsymbol{\eta})\,, \tag{46}$$

with equality if $(\mathbf{x},\mathbf{y}) \in H^n = \{(1,0),(0,1),(0,0)\}^n$, because the elements of the sum are $0\cdot f(\boldsymbol{\eta})$ for all $\boldsymbol{\eta} \in H^n \setminus \{(\mathbf{x},\mathbf{y})\}$ and $1\cdot f(\boldsymbol{\eta})$ only for $\boldsymbol{\eta} = (\mathbf{x},\mathbf{y})$.

Given that realizations of random variable (X, Y) are independent and that for a given $\eta_i \in H$, only one of the three products is computed[19], we get

$$\begin{aligned}
\mathbf{E}\,[f(\mathbf{X},\mathbf{Y})] &\leq \mathbf{E}\left[\sum_{\boldsymbol{\eta}\in H^n}\left[\left(\prod_{i:\eta_i=(1,0)} X_i\right)\left(\prod_{i:\eta_i=(0,1)} Y_i\right)\left(\prod_{i:\eta_i=(0,0)} 1-X_i-Y_i\right)\right]\cdot f(\boldsymbol{\eta})\right] \\
&= \sum_{\boldsymbol{\eta}\in H^n}\mathbf{E}\left[\left(\prod_{i:\eta_i=(1,0)} X_i\right)\left(\prod_{i:\eta_i=(0,1)} Y_i\right)\left(\prod_{i:\eta_i=(0,0)} 1-X_i-Y_i\right)\right]\cdot f(\boldsymbol{\eta}) \\
&= \sum_{\boldsymbol{\eta}\in H^n}\left[\left(\prod_{i:\eta_i=(1,0)} \mathbf{E}(X_i)\right)\left(\prod_{i:\eta_i=(0,1)} \mathbf{E}(Y_i)\right)\left(\prod_{i:\eta_i=(0,0)} 1-\mathbf{E}(X_i)-\mathbf{E}(Y_i)\right)\right]\cdot f(\boldsymbol{\eta}) \\
&= \sum_{\boldsymbol{\eta}\in H^n}\left[\left(\prod_{i:\eta_i=(1,0)} \mu_X\right)\left(\prod_{i:\eta_i=(0,1)} \mu_Y\right)\left(\prod_{i:\eta_i=(0,0)} 1-\mu_X-\mu_Y\right)\right]\cdot f(\boldsymbol{\eta})\,.
\end{aligned}$$

This becomes an equality when $(\mathbf{X}, \mathbf{Y})$ takes values in H^n (as we explain after equation 46). We therefore conclude that $\mathbf{E}\big[f(\mathbf{X},\mathbf{Y})\big] \;\leq\; \mathbf{E}\big[f(\mathbf{X}',\mathbf{Y}')\big]$. ■

Lemma 53 (Proof of Equation 45) *Consider $H = \{(1,0),(0,1),(0,0)\}$ and an integer $n > 0$. Any point $(\mathbf{x},\mathbf{y}) \in \big([0,1]\times[0,1]\big)^n$ can be written as a convex combination of the extreme points $\boldsymbol{\eta} = (\eta_1, \eta_2, \ldots, \eta_n) \in H^n$:*

$$(\mathbf{x},\mathbf{y}) \;=\; \sum_{\boldsymbol{\eta}\in H^n} \rho_{\boldsymbol{\eta}}(\mathbf{x},\mathbf{y})\cdot\boldsymbol{\eta}\,,$$

where

$$\rho_{\boldsymbol{\eta}}(\mathbf{x},\mathbf{y}) \;\stackrel{\text{def}}{=}\; \left(\prod_{i:\eta_i=(1,0)} x_i\right)\left(\prod_{i:\eta_i=(0,1)} y_i\right)\left(\prod_{i:\eta_i=(0,0)} 1-x_i-y_i\right)\,.$$

Proof We prove the result by induction over vector size n.
Proof for $n = 1$:

$$\begin{aligned}
\sum_{\boldsymbol{\eta}\in H}\rho_{\boldsymbol{\eta}}((x_1,y_1))\cdot\boldsymbol{\eta} &= x_1\cdot((1,0)) + y_1\cdot((0,1)) + (1-x_1-y_1)\cdot((0,0)) \\
&= ((x_1,y_1))\,.
\end{aligned}$$

Proof for $n > 1$: We suppose that the result is true for any vector $(\mathbf{x},\mathbf{y})$ of a particular size n (this is our induction hypothesis) and we prove that it implies

$$\sum_{(\boldsymbol{\eta},\eta_{n+1})\in H^{n+1}}\Big[\rho_{(\boldsymbol{\eta},\eta_{n+1})}\big((\mathbf{x},\mathbf{y}),(x_{n+1},y_{n+1})\big)\Big]\cdot(\boldsymbol{\eta},\eta_{n+1}) \;=\; \big((\mathbf{x},\mathbf{y}),\,(x_{n+1},y_{n+1})\big)\,,$$

where $(\mathbf{a},\, b)$ denotes a vector $\mathbf{a}$, augmented by one element b.

19. The equality between the second and third lines follows from the fact that each expectation inside the sum of Line 2 can be rewritten as the following product of independent random variables:

$$\mathbf{E}\Big[\prod_{\eta_i} g_{\eta_i}(X_i,Y_i)\Big] \quad\text{with}\quad g_{\eta_i}(X_i,Y_i) \;\stackrel{\text{def}}{=}\; \begin{cases} X_i & \text{if } \eta_i = (1,0) \\ Y_i & \text{if } \eta_i = (0,1) \\ 1-X_i-Y_i & \text{otherwise.}\end{cases}$$

We have

$$\begin{aligned}
&\sum_{(\boldsymbol{\eta},\eta_{n+1})\in H^{n+1}} \Big[\rho_{(\boldsymbol{\eta},\eta_{n+1})}\big((\mathbf{x},\mathbf{y}),(x_{n+1},y_{n+1})\big)\Big]\cdot(\boldsymbol{\eta},\eta_{n+1}) \\
&= \sum_{\boldsymbol{\eta}\in H^n} \rho_{\boldsymbol{\eta}}(\mathbf{x},\mathbf{y})\cdot x_{n+1}\cdot\big(\boldsymbol{\eta},(1,0)\big) + \sum_{\boldsymbol{\eta}\in H^n} \rho_{\boldsymbol{\eta}}(\mathbf{x},\mathbf{y})\cdot y_{n+1}\cdot\big(\boldsymbol{\eta},(0,1)\big) \\
&\qquad + \sum_{\boldsymbol{\eta}\in H^n} \rho_{\boldsymbol{\eta}}(\mathbf{x},\mathbf{y})\cdot(1-x_{n+1}-y_{n+1})\cdot\big(\boldsymbol{\eta},(0,0)\big) \\
&= \left(\sum_{\boldsymbol{\eta}\in H^n} \rho_{\boldsymbol{\eta}}(\mathbf{x},\mathbf{y})\cdot(x_{n+1}+y_{n+1}+1-x_{n+1}-y_{n+1})\cdot\boldsymbol{\eta},\ \sum_{\boldsymbol{\eta}\in H^n} \rho_{\boldsymbol{\eta}}(\mathbf{x},\mathbf{y})\cdot\big(x_{n+1},y_{n+1}\big)\right) \\
&= \left(\sum_{\boldsymbol{\eta}\in H^n} \rho_{\boldsymbol{\eta}}(\mathbf{x},\mathbf{y})\cdot\boldsymbol{\eta},\ \sum_{\boldsymbol{\eta}\in H^n} \rho_{\boldsymbol{\eta}}(\mathbf{x},\mathbf{y})\cdot\big(x_{n+1},y_{n+1}\big)\right) \\
&= \Big((\mathbf{x},\mathbf{y}),\ \big(x_{n+1},y_{n+1}\big)\Big)\,.
\end{aligned}$$

For the last equality, the $(\mathbf{x},\mathbf{y})$ term of the vector above is obtained from the induction hypothesis and the last couple is a direct consequence of the following equality:

$$\sum_{\boldsymbol{\eta}\in H^n} \rho_{\boldsymbol{\eta}}(\mathbf{x},\mathbf{y}) \;=\; \prod_{i=1}^{n}\Big(x_i+y_i+1-x_i-y_i\Big) \;=\; 1\,.$$

■

Proposition 54 (Concavity of Equation 36) *The function $F_{\mathcal{C}}(d,e)$ is concave.*

Proof We show that the Hessian matrix of $F_{\mathcal{C}}(d,e)$ is a negative semi-definite matrix. In other words, we need to prove that

$$\frac{\partial^2 F_{\mathcal{C}}(d,e)}{\partial d^2}\le 0\,;\quad \frac{\partial^2 F_{\mathcal{C}}(d,e)}{\partial e^2}\le 0\,;\quad \frac{\partial^2 F_{\mathcal{C}}(d,e)}{\partial d^2}\frac{\partial^2 F_{\mathcal{C}}(d,e)}{\partial e^2}-\left(\frac{\partial^2 F_{\mathcal{C}}(d,e)}{\partial d\partial e}\right)^2\ge 0\,.$$

Indeed, we have

$$\begin{aligned}
\frac{\partial^2 F_{\mathcal{C}}(d,e)}{\partial d^2} &= \frac{2(1-4e)^2}{(2d-1)^3}\le 0 \quad \forall e\in[0,1], d\in\Big[0,\frac{1}{2}\Big]\,,\\
\frac{\partial^2 F_{\mathcal{C}}(d,e)}{\partial e^2} &= \frac{8}{2d-1}\le 0 \qquad \forall e\in[0,1], d\in\Big[0,\frac{1}{2}\Big]\,,\\
\frac{\partial^2 F_{\mathcal{C}}(d,e)}{\partial d^2}\frac{\partial^2 F_{\mathcal{C}}(d,e)}{\partial e^2}-\left(\frac{\partial^2 F_{\mathcal{C}}(d,e)}{\partial d\partial e}\right)^2 &= \frac{2(1-4e)^2}{(2d-1)^3}\cdot\frac{8}{2d-1}-\left(\frac{4-16e}{(1-2d)^2}\right)^2=0\,.
\end{aligned}$$

■

Appendix B. A General PAC-Bayesian Theorem for Tuples of Voters and Aligned Posteriors

This section presents a change of measure inequality that generalizes both Lemmas 30 and 34, and a PAC-Bayesian theorem that generalizes both Theorems 31 and 35. As these generalizations require more complex notation and ideas, it is provided as an appendix and the simpler versions of the main paper have separate proofs.

Let $\mathcal{H}$ be a countable self-complemented set real-valued functions. In the general setting, we recall that $\mathcal{H}$ is self-complemented if there exists a bijection $c : \mathcal{H} \to \mathcal{H}$ such that $c(f) = -f$ for any $f \in \mathcal{H}$. Moreover, for a distribution Q aligned on a prior distribution P and for any $f \in \mathcal{H}$, we have

$$Q(f) + Q(c(f)) \quad = \quad P(f) + P(c(f))\,.$$

First, we need to define the following notation. Let $\mathbf{k}$ be a sequence of length k, containing numbers representing indices of voters. Let $f_{\mathbf{k}} : \mathcal{X} \to \overline{\mathcal{Y}}^k$ be a function that outputs a tuple of votes, such that $f_{\mathbf{k}}(x) \stackrel{\text{def}}{=} \langle f_{\mathbf{k}_1}(x), \ldots, f_{\mathbf{k}_k}(x)\rangle\,.$

Let us recall that P^k and Q^k are Cartesian products of probability distributions P and Q. Thus, the probability of drawing $f_{\mathbf{k}} \sim Q^k$ is given by

$$Q^k(f_{\mathbf{k}}) \stackrel{\text{def}}{=} Q(f_{\mathbf{k}_1}) \cdot Q(f_{\mathbf{k}_2}) \cdot \ldots \cdot Q(f_{\mathbf{k}_k}) = \prod_{i=1}^{k} Q(f_{\mathbf{k}_i})\,.$$

Finally, for each $f_{\mathbf{k}}$ and each $j \in \{0, \ldots, 2^k - 1\}$, let

$$f^{[j]}_{\mathbf{k}}(x) \stackrel{\text{def}}{=} \langle f^{(s^j_1)}_{\mathbf{k}_1}(x), \ldots, f^{(s^j_k)}_{\mathbf{k}_k}(x)\rangle\ ,$$

where $s^j_1 s^j_2 ... s^j_k$ is the binary representation of the number j, and where $f^{(0)} = f$ and $f^{(1)} = c(f)$. Note that $f^{[0]}_{\mathbf{k}} = f_{\mathbf{k}}$.

To prove the next PAC-Bayesian theorem, we make use of the following change of measure inequality.

Theorem 55 (Change of measure inequality for tuples of voters and aligned posteriors) *For any self-complemented set $\mathcal{H}$, for any distribution P on $\mathcal{H}$, for any distribution Q aligned on P, and for any measurable function $\phi : \mathcal{H}^k \to \mathbb{R}$ for which $\phi(f^{[j]}_{\mathbf{k}}) = \phi(f^{[j']}_{\mathbf{k}})$ for any $j, j' \in \{0, \ldots, 2^k - 1\}$, we have*

$$\mathop{\mathbf{E}}_{f_{\mathbf{k}} \sim Q^k} \phi(f_{\mathbf{k}}) \ \leq \ \ln\left(\mathop{\mathbf{E}}_{f_{\mathbf{k}} \sim P^k} e^{\phi(f_{\mathbf{k}})}\right).$$

Proof First, note that one can change the expectation over Q^k to an expectation over P^k, using the fact that $\phi(f_{\mathbf{k}}^{[j]}) = \phi(f_{\mathbf{k}}^{[j']})$ for any $j, j' \in \{0, \ldots, 2^k - 1\}$ and that Q is aligned on P.

$$
\begin{aligned}
&2^k \cdot \mathop{\mathbf{E}}_{f_{\mathbf{k}} \sim Q^k} \phi(f_{\mathbf{k}}) \\
&= \int_{\mathcal{H}^k} df_{\mathbf{k}}\; Q^k(f_{\mathbf{k}}^{[0]})\, \phi(f_{\mathbf{k}}^{[0]}) + \int_{\mathcal{H}^k} df_{\mathbf{k}}\; Q^k(f_{\mathbf{k}}^{[1]})\, \phi(f_{\mathbf{k}}^{[1]}) + \ldots + \int_{\mathcal{H}^k} df_{\mathbf{k}}\; Q^k(f_{\mathbf{k}}^{[2^k-1]})\, \phi(f_{\mathbf{k}}^{[2^k-1]}) \\
&= \int_{\mathcal{H}^k} df_{\mathbf{k}}\; Q^k(f_{\mathbf{k}}^{[0]})\, \phi(f_{\mathbf{k}}) \;+\; \int_{\mathcal{H}^k} df_{\mathbf{k}}\; Q^k(f_{\mathbf{k}}^{[1]})\, \phi(f_{\mathbf{k}}) \;+\; \ldots \;+\; \int_{\mathcal{H}^k} df_{\mathbf{k}}\; Q^k(f_{\mathbf{k}}^{[2^k-1]})\, \phi(f_{\mathbf{k}}) \\
&= \int_{\mathcal{H}^k} df_{\mathbf{k}}\; \sum_{j=0}^{2^k-1} \left(Q^k(f_{\mathbf{k}}^{[j]}) \right) \phi(f_{\mathbf{k}}) \\
&= \int_{\mathcal{H}^k} df_{\mathbf{k}}\; \sum_{j=0}^{2^k-1} \left(\prod_{i=1}^{k} \left[Q(f_{\mathbf{k}_i}^{(s_i^j)}) \right] \right) \phi(f_{\mathbf{k}}) \qquad (47) \\
&= \int_{\mathcal{H}^k} df_{\mathbf{k}}\; \prod_{i=1}^{k} \left[Q(f_{\mathbf{k}_i}^{(0)}) + Q(f_{\mathbf{k}_i}^{(1)}) \right] \phi(f_{\mathbf{k}}) \qquad (48) \\
&= \int_{\mathcal{H}^k} df_{\mathbf{k}}\; \prod_{i=1}^{k} \left[Q(f_{\mathbf{k}_i}) + Q(c(f_{\mathbf{k}_i})) \right] \phi(f_{\mathbf{k}}) \\
&= \int_{\mathcal{H}^k} df_{\mathbf{k}}\; \prod_{i=1}^{k} \left[P(f_{\mathbf{k}_i}) + P(c(f_{\mathbf{k}_i})) \right] \phi(f_{\mathbf{k}}) \\
&\vdots \\
&= 2^k \cdot \mathop{\mathbf{E}}_{f_{\mathbf{k}} \sim P^k} \phi(f_{\mathbf{k}})\,,
\end{aligned}
$$

where we obtain Line (48) from Line (47) by developing the terms of the product of Line (48).

The result is obtained by changing the expectation over Q^k to an expectation over P^k, and then by applying Jensen's inequality (Lemma 47, in Appendix A).

$$
\mathop{\mathbf{E}}_{f_{\mathbf{k}} \sim Q^k} \phi(f_{\mathbf{k}}) \;=\; \mathop{\mathbf{E}}_{f_{\mathbf{k}} \sim P^k} \phi(f_{\mathbf{k}}) \;=\; \mathop{\mathbf{E}}_{f_{\mathbf{k}} \sim P^k} \ln e^{\phi(f_{\mathbf{k}})} \;\leq\; \ln \left(\mathop{\mathbf{E}}_{f_{\mathbf{k}} \sim P^k} e^{\phi(f_{\mathbf{k}})} \right).
$$

■

Theorem 56 (General PAC-Bayesian theorem for tuples of voters and aligned posteriors) *For any distribution D on $\mathcal{X}\times\mathcal{Y}$, any self-complemented set $\mathcal{H}$ of voters $\mathcal{X}\to\overline{\mathcal{Y}}$, any prior distribution P on $\mathcal{H}$, any integer $k\geq 1$, any convex function $\mathcal{D}:[0,1]\times[0,1]\to\mathbb{R}$ and loss function $\mathcal{L}:\overline{\mathcal{Y}}^k\times\mathcal{Y}^k\to[0,1]$ for which $\mathcal{D}\left(\mathbb{E}_S^{\mathcal{L}}(f_{\mathbf{k}}^{[j]}),\,\mathbb{E}_D^{\mathcal{L}}(f_{\mathbf{k}}^{[j]})\right)=\mathcal{D}\left(\mathbb{E}_S^{\mathcal{L}}(f_{\mathbf{k}}^{[j']}),\,\mathbb{E}_D^{\mathcal{L}}(f_{\mathbf{k}}^{[j']})\right)$, for any $j,j'\in\{0,\ldots,2^k-1\}$, for any $m'>0$ and any $\delta\in(0,1]$, we have*

$$\Pr_{S\sim D^m}\left(\begin{array}{l}\textit{For all posteriors } Q \textit{ aligned on } P:\\ \mathcal{D}\Big(\mathop{\mathbf{E}}_{f_{\mathbf{k}}\sim Q^k}\mathbb{E}_S^{\mathcal{L}}(f_{\mathbf{k}}),\mathop{\mathbf{E}}_{f_{\mathbf{k}}\sim Q^k}\mathbb{E}_D^{\mathcal{L}}(f_{\mathbf{k}})\Big)\leq\dfrac{1}{m'}\left[\ln\left(\dfrac{1}{\delta}\mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{f_{\mathbf{k}}\sim P^k}e^{\,m'\cdot\mathcal{D}\left(\mathbb{E}_S^{\mathcal{L}}(f_{\mathbf{k}}),\,\mathbb{E}_D^{\mathcal{L}}(f_{\mathbf{k}})\right)}\right)\right]\end{array}\right)\geq 1-\delta\,.$$

Proof This proof follows most of the steps of Theorem 18.
We have that $\mathop{\mathbf{E}}_{f_{\mathbf{k}}\sim P^k}e^{\,m'\cdot\mathcal{D}\left(\mathbb{E}_S^{\mathcal{L}}(f_{\mathbf{k}}),\,\mathbb{E}_D^{\mathcal{L}}(f_{\mathbf{k}})\right)}$ is a non-negative random variable. By Markov's inequality, we have

$$\Pr_{S\sim D^m}\left(\mathop{\mathbf{E}}_{f_{\mathbf{k}}\sim P^k}e^{\,m'\cdot\mathcal{D}\left(\mathbb{E}_S^{\mathcal{L}}(f_{\mathbf{k}}),\,\mathbb{E}_D^{\mathcal{L}}(f_{\mathbf{k}})\right)}\leq\frac{1}{\delta}\mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{f_{\mathbf{k}}\sim P^k}e^{\,m'\cdot\mathcal{D}\left(\mathbb{E}_S^{\mathcal{L}}(f_{\mathbf{k}}),\,\mathbb{E}_D^{\mathcal{L}}(f_{\mathbf{k}})\right)}\right)\geq 1-\delta\,.$$

Hence, by taking the logarithm on each side of the innermost inequality, we obtain

$$\Pr_{S\sim D^m}\left(\ln\left[\mathop{\mathbf{E}}_{f_{\mathbf{k}}\sim P^k}e^{\,m'\cdot\mathcal{D}\left(\mathbb{E}_S^{\mathcal{L}}(f_{\mathbf{k}}),\,\mathbb{E}_D^{\mathcal{L}}(f_{\mathbf{k}})\right)}\right]\leq\ln\left[\frac{1}{\delta}\mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{f_{\mathbf{k}}\sim P^k}e^{\,m'\cdot\mathcal{D}\left(\mathbb{E}_S^{\mathcal{L}}(f_{\mathbf{k}}),\,\mathbb{E}_D^{\mathcal{L}}(f_{\mathbf{k}})\right)}\right]\right)\geq 1-\delta\,.$$

Now, instead of using the change of measure inequality of Lemma 17, we use the change of measure inequality of Theorem 55 on the left side of innermost inequality, with $\phi(f_{\mathbf{k}})=m'\cdot\mathcal{D}\left(\mathbb{E}_S^{\mathcal{L}}(f_{\mathbf{k}}),\,\mathbb{E}_D^{\mathcal{L}}(f_{\mathbf{k}})\right)$. We then use Jensen's inequality (Lemma 47, in Appendix A), exploiting the convexity of $\mathcal{D}$.

$$\begin{aligned}\forall\,Q\text{ aligned on }P:\ \ln\left[\mathop{\mathbf{E}}_{f_{\mathbf{k}}\sim P^k}e^{\,m'\cdot\mathcal{D}(\mathbb{E}_S^{\mathcal{L}}(f_{\mathbf{k}}),\,\mathbb{E}_D^{\mathcal{L}}(f_{\mathbf{k}}))}\right]&\geq m'\cdot\mathop{\mathbf{E}}_{f_{\mathbf{k}}\sim Q^k}\mathcal{D}(\mathbb{E}_S^{\mathcal{L}}(f_{\mathbf{k}}),\mathbb{E}_D^{\mathcal{L}}(f_{\mathbf{k}}))\\&\geq m'\cdot\mathcal{D}(\mathop{\mathbf{E}}_{f_{\mathbf{k}}\sim Q^k}\mathbb{E}_S^{\mathcal{L}}(f_{\mathbf{k}}),\mathop{\mathbf{E}}_{f_{\mathbf{k}}\sim Q^k}\mathbb{E}_D^{\mathcal{L}}(f_{\mathbf{k}}))\,.\end{aligned}$$

We therefore have

$$\Pr_{S\sim D^m}\left(\begin{array}{l}\text{For all posteriors } Q \text{ aligned on } P:\\ m'\cdot\mathcal{D}(\mathop{\mathbf{E}}_{f_{\mathbf{k}}\sim Q^k}\mathbb{E}_S^{\mathcal{L}}(f_{\mathbf{k}}),\mathop{\mathbf{E}}_{f_{\mathbf{k}}\sim Q^k}\mathbb{E}_D^{\mathcal{L}}(f_{\mathbf{k}}))\leq\ln\left[\frac{1}{\delta}\mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{f_{\mathbf{k}}\sim P^k}e^{\,m'\cdot\mathcal{D}(\mathbb{E}_S^{\mathcal{L}}(f_{\mathbf{k}}),\,\mathbb{E}_D^{\mathcal{L}}(f_{\mathbf{k}}))}\right]\end{array}\right)\geq 1-\delta\,.$$

The result then follows from easy calculations. ■

References

Arindam Banerjee. On Bayesian bounds. In *ICML*, pages 81–88, 2006.

Luc Bégin, Pascal Germain, François Laviolette, and Jean-Francis Roy. PAC-Bayesian theory for transductive learning. In *AISTATS*, pages 105–113, 2014.

C.L. Blake and C.J. Merz. *UCI Repository of Machine Learning Databases*. Department of Information and Computer Science, Irvine, CA: University of California, http://www.ics.uci.edu/∼mlearn/MLRepository.html, 1998.

John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Annual Meeting-Association for Computational Linguistics*, volume 45, page 440, 2007.

Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

Olivier Catoni. *PAC-Bayesian Supervised Classification: The Thermodynamics of Statistical Learning*. Monograph series of the Institute of Mathematical Statistics, http://arxiv.org/abs/0712.0248, 2007.

Minmin Chen, Kilian Q. Weinberger, and John Blitzer. Co-training for domain adaptation. In *NIPS*, pages 2456–2464, 2011.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3): 273–297, 1995.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*, chapter 12. Wiley, 1991.

Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge University Press, Cambridge, U.K., 2000.

Joachim Dahl and Lieven Vandenberghe. CVXOPT, 2007. `http://mloss.org/software/view/34/`.

Amit Daniely, Sivan Sabato, Shai Ben-David, and Shai Shalev-Shwartz. Multiclass learnability and the ERM principle. *CoRR*, abs/1308.2893, 2013.

Philip Derbeko, Ran El-Yaniv, and Ron Meir. Explicit learning curves for transduction and application to clustering and compression algorithms. *J. Artif. Intell. Res. (JAIR)*, 22: 117–142, 2004.

Mark Dredze, Alex Kulesza, and Koby Crammer. Multi-domain learning by confidence-weighted parameter combination. *Machine Learning*, 79(1-2):123–149, 2010.

Mahdi Milani Fard and Joelle Pineau. PAC-Bayesian model selection for reinforcement learning. In *NIPS*, pages 1624–1632, 2010.

Mahdi Milani Fard, Joelle Pineau, and Csaba Szepesvári. PAC-Bayesian policy evaluation for reinforcement learning. In *UAI*, pages 195–202, 2011.

Sally Floyd and Manfred Warmuth. Sample compression, learnability, and the Vapnik-Chervonenkis dimension. *Machine Learning*, 21(3):269–304, 1995.

A. Gelman, J.B. Carlin, H.S. Stern, and D.B. Rubin. *Bayesian Data Analysis*. Chapman & Hall/CRC, 2004. ISBN 9781584883883.

Pascal Germain, Alexandre Lacasse, François Laviolette, and Mario Marchand. PAC-Bayesian learning of linear classifiers. In *ICML*, page 45, 2009.

Pascal Germain, Alexandre Lacoste, François Laviolette, Mario Marchand, and Sara Shanian. A PAC-Bayes sample-compression approach to kernel methods. In *ICML*, pages 297–304, 2011.

Pascal Germain, Amaury Habrard, François Laviolette, and Emilie Morvant. A PAC-Bayesian approach for domain adaptation with specialization to linear classifiers. In *ICML (3)*, pages 738–746, 2013.

Sébastien Giguère, François Laviolette, Mario Marchand, and Khadidja Sylla. Risk bounds and learning algorithms for the regression approach to structured output prediction. In *ICML (1)*, pages 107–114, 2013.

Matthew Higgs and John Shawe-Taylor. A PAC-Bayes bound for tailored density estimation. In *ALT*, pages 148–162, 2010.

Alexandre Lacasse, François Laviolette, Mario Marchand, Pascal Germain, and Nicolas Usunier. PAC-Bayes bounds for the risk of the majority vote and the variance of the Gibbs classifier. In *NIPS*, pages 769–776, 2006.

Alexandre Lacoste, François Laviolette, and Mario Marchand. Bayesian comparison of machine learning algorithms on single and multiple datasets. In *AISTATS*, pages 665–675, 2012.

John Langford. Tutorial on practical prediction theory for classification. *Journal of Machine Learning Research*, 6:273–306, 2005.

John Langford and Matthias Seeger. Bounds for averaging classifiers. Technical report, Carnegie Mellon, Departement of Computer Science, 2001.

John Langford and John Shawe-Taylor. PAC-Bayes & margins. In *NIPS*, pages 423–430, 2002.

François Laviolette and Mario Marchand. PAC-Bayes risk bounds for sample-compressed Gibbs classifiers. In *ICML*, pages 481–488, 2005.

François Laviolette and Mario Marchand. PAC-Bayes risk bounds for stochastic averages and majority votes of sample-compressed classifiers. *Journal of Machine Learning Research*, 8:1461–1487, 2007.

François Laviolette, Mario Marchand, and Jean-Francis Roy. From PAC-Bayes bounds to quadratic programs for majority votes. In *ICML*, pages 649–656, 2011.

Yann Lecun and Corinna Cortes. The MNIST database of handwritten digits. URL `http://yann.lecun.com/exdb/mnist/`.

Guy Lever, François Laviolette, and John Shawe-Taylor. Distribution-dependent PAC-Bayes priors. In *ALT*, pages 119–133, 2010.

Guy Lever, François Laviolette, and John Shawe-Taylor. Tighter PAC-Bayes bounds through distribution-dependent priors. *Theoretical Computer Science*, 473:4–28, 2013.

Ben London, Bert Huang, Benjamin Taskar, and Lise Getoor. PAC-Bayesian collective stability. In *AISTATS*, pages 585–594, 2014.

Andreas Maurer. A note on the PAC-Bayesian theorem. *CoRR*, cs.LG/0411099, 2004.

David McAllester. Some PAC-Bayesian theorems. *Machine Learning*, 37(3):355–363, 1999.

David McAllester. PAC-Bayesian stochastic model selection. *Machine Learning*, 51(1):5–21, 2003a.

David McAllester. Simplified PAC-Bayesian margin bounds. In *COLT*, pages 203–215, 2003b.

David McAllester. Generalization bounds and consistency for structured labeling. In Gökhan Bakır, Thomas Hofmann, Bernhard Schölkopf, Alexander J. Smola, Ben Taskar, and S. V. N. Vishwanathan, editors, *Predicting Structured Data*, chapter 11, pages 247–261. MIT Press, Cambridge, MA, 2007.

David McAllester. A PAC-Bayesian tutorial with a dropout bound. *CoRR*, abs/1307.2118, 2013.

W. Mendenhall. Nonparametric statistics. *Introduction to Probability and Statistics*, 604, 1983.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python . *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Liva Ralaivola, Marie Szafranski, and Guillaume Stempfel. Chromatic PAC-Bayes bounds for non-iid data: Applications to ranking and stationary β-mixing processes. *Journal of Machine Learning Research*, 11:1927–1956, 2010.

Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1988.

Robert E. Schapire and Yoram Singer. Improved boosting using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999.

Robert E. Schapire, Yoav Freund, Peter Bartlett, and Wee Sun Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26: 1651–1686, 1998.

Bernhard Schölkopf, Ralf Herbrich, and Alex J. Smola. A generalized representer theorem. In *COLT/EuroCOLT*, pages 416–426, 2001.

Matthias Seeger. PAC-Bayesian generalization bounds for Gaussian processes. *Journal of Machine Learning Research*, 3:233–269, 2002.

Matthias Seeger. *Bayesian Gaussian Process Models: PAC-Bayesian Generalisation Error Bounds and Sparse Approximations*. PhD thesis, University of Edinburgh, 2003.

Yevgeny Seldin and Naftali Tishby. PAC-Bayesian generalization bound for density estimation with application to co-clustering. In *AISTATS*, pages 472–479, 2009.

Yevgeny Seldin and Naftali Tishby. PAC-Bayesian analysis of co-clustering and beyond. *Journal of Machine Learning Research*, 11:3595–3646, 2010.

Yevgeny Seldin, Peter Auer, François Laviolette, John Shawe-Taylor, and Ronald Ortner. PAC-Bayesian analysis of contextual bandits. In *NIPS*, pages 1683–1691, 2011.

Yevgeny Seldin, François Laviolette, Nicolò Cesa-Bianchi, John Shawe-Taylor, and Peter Auer. PAC-Bayesian inequalities for martingales. *IEEE Transactions on Information Theory*, 58(12):7086–7093, 2012.

Chunhua Shen and Hanxi Li. Boosting through optimization of margin distributions. *IEEE Transactions on Neural Networks*, 21(4):659–666, 2010.

Ilya O. Tolstikhin and Yevgeny Seldin. PAC-Bayes-empirical-bernstein inequality. In *NIPS*, pages 109–117, 2013.

Malik Younsi. Proof of a combinatorial conjecture coming from the PAC-Bayesian machine learning theory. *ArXiv E-Prints*, 2012. URL `http://arxiv.org/abs/1209.0824v1`.

Journal of Machine Learning Research 16 (2015) 861-911 Submitted 1/14; Revised 10/14; Published 4/15

A Statistical Perspective on Algorithmic Leveraging

Ping Ma PINGMA@UGA.EDU
Department of Statistics
University of Georgia
Athens, GA 30602

Michael W. Mahoney MMAHONEY@STAT.BERKELEY.EDU
International Computer Science Institute and Department of Statistics
University of California at Berkeley
Berkeley, CA 94720

Bin Yu BINYU@STAT.BERKELEY.EDU
Department of Statistics
University of California at Berkeley
Berkeley, CA 94720

Editor: Alexander Rakhlin

Abstract

One popular method for dealing with large-scale data sets is sampling. For example, by using the empirical statistical leverage scores as an importance sampling distribution, the method of *algorithmic leveraging* samples and rescales rows/columns of data matrices to reduce the data size before performing computations on the subproblem. This method has been successful in improving computational efficiency of algorithms for matrix problems such as least-squares approximation, least absolute deviations approximation, and low-rank matrix approximation. Existing work has focused on algorithmic issues such as worst-case running times and numerical issues associated with providing high-quality implementations, but none of it addresses statistical aspects of this method.

In this paper, we provide a simple yet effective framework to evaluate the statistical properties of algorithmic leveraging in the context of estimating parameters in a linear regression model with a fixed number of predictors. In particular, for several versions of leverage-based sampling, we derive results for the bias and variance, both conditional and unconditional on the observed data. We show that from the statistical perspective of bias and variance, neither leverage-based sampling nor uniform sampling dominates the other. This result is particularly striking, given the well-known result that, from the algorithmic perspective of worst-case analysis, leverage-based sampling provides uniformly superior worst-case algorithmic results, when compared with uniform sampling.

Based on these theoretical results, we propose and analyze two new leveraging algorithms: one constructs a smaller least-squares problem with "shrinkage" leverage scores (SLEV), and the other solves a smaller and unweighted (or biased) least-squares problem (LEVUNW). A detailed empirical evaluation of existing leverage-based methods as well as these two new methods is carried out on both synthetic and real data sets. The empirical results indicate that our theory is a good predictor of practical performance of existing and new leverage-based algorithms and that the new algorithms achieve improved performance. For example, with the same computation reduction as in the original algorithmic leveraging approach, our proposed SLEV typically leads to improved biases and variances both

unconditionally and conditionally (on the observed data), and our proposed LEVUNW typically yields improved unconditional biases and variances.

Keywords: randomized algorithm, leverage scores, subsampling, least squares, linear regression

1. Introduction

One popular method for dealing with large-scale data sets is sampling. In this approach, one first chooses a small portion of the full data, and then one uses this sample as a surrogate to carry out computations of interest for the full data. For example, one might randomly sample a small number of rows from an input matrix and use those rows to construct a low-rank approximation to the original matrix, or one might randomly sample a small number of constraints or variables in a regression problem and then perform a regression computation on the subproblem thereby defined. For many problems, it is very easy to construct "worst-case" input for which *uniform* random sampling will perform very poorly. Motivated by this, there has been a great deal of work on developing algorithms for matrix-based machine learning and data analysis problems that construct the random sample in a *nonuniform* data-dependent fashion (Mahoney, 2011).

Of particular interest here is when that data-dependent sampling process selects rows or columns from the input matrix according to a probability distribution that depends on the empirical statistical leverage scores of that matrix. This recently-developed approach of *algorithmic leveraging* has been applied to matrix-based problems that are of interest in large-scale data analysis, e.g., least-squares approximation (Drineas et al., 2006, 2010), least absolute deviations regression (Clarkson et al., 2013; Meng and Mahoney, 2013), and low-rank matrix approximation (Mahoney and Drineas, 2009; Clarkson and Woodruff, 2013). Typically, the leverage scores are computed approximately (Drineas et al., 2012; Clarkson et al., 2013), or otherwise a random projection (Ailon and Chazelle, 2010; Clarkson et al., 2013) is used to precondition by approximately uniformizing them (Drineas et al., 2010; Avron et al., 2010; Meng et al., 2014). A detailed discussion of this approach can be found in the recent review monograph on randomized algorithms for matrices and matrix-based data problems (Mahoney, 2011).

This algorithmic leveraging paradigm has already yielded impressive algorithmic benefits: by preconditioning with a high-quality numerical implementation of a Hadamard-based random projection, the Blendenpik code of Avron et al. (2010) "beats Lapack's[1] direct dense least-squares solver by a large margin on essentially any dense tall matrix;" the LSRN algorithm of Meng et al. (2014) preconditions with a high-quality numerical implementation of a normal random projection in order to solve large over-constrained least-squares problems on clusters with high communication cost, e.g., on Amazon Elastic Cloud Compute clusters; the solution to the ℓ_1 regression or least absolute deviations problem as well as to quantile regression problems can be approximated for problems with billions of constraints (Clarkson et al., 2013; Yang et al., 2013); and CUR-based low-rank matrix approximations (Mahoney and Drineas, 2009) have been used for structure extraction in DNA SNP matrices of size thousands of individuals by hundreds of thousands of

1. Lapack (short for Linear Algebra PACKage) is a high-quality and widely-used software library of numerical routines for solving a wide range of numerical linear algebra problems.

SNPs (Paschou et al., 2007, 2010). In spite of these impressive *algorithmic* results, none of this recent work on leveraging or leverage-based sampling addresses *statistical* aspects of this approach. This is in spite of the central role of statistical leverage, a traditional concept from regression diagnostics (Hoaglin and Welsch, 1978; Chatterjee and Hadi, 1986; Velleman and Welsch, 1981).

In this paper, we bridge that gap by providing the first statistical analysis of the algorithmic leveraging paradigm. We do so in the context of parameter estimation in fitting linear regression models for large-scale data—where, by "large-scale," we mean that the data define a high-dimensional problem in terms of sample size n, as opposed to the dimension p of the parameter space. Although $n \gg p$ is the classical regime in theoretical statistics, it is a relatively new phenomenon that in practice we routinely see a sample size n in the hundreds of thousands or millions or more. This is a size regime where sampling methods such as algorithmic leveraging are indispensable to meet computational constraints.

Our main theoretical contribution is to provide an analytic framework for evaluating the statistical properties of algorithmic leveraging. This involves performing a Taylor series analysis around the ordinary least-squares solution to approximate the subsampling estimators as linear combinations of random sampling matrices. Within this framework, we consider biases and variances, both conditioned as well as not conditioned on the data, for several versions of the basic algorithmic leveraging procedure. We show that both leverage-based sampling and uniform sampling are unbiased to leading order; and that while leverage-based sampling improves the "size-scale" of the variance, relative to uniform sampling, the presence of very small leverage scores can inflate the variance considerably. It is well-known that, from the algorithmic perspective of worst-case analysis, leverage-based sampling provides uniformly superior worst-case algorithmic results, when compared with uniform sampling. However, our statistical analysis here reveals that from the statistical perspective of bias and variance, neither leverage-based sampling nor uniform sampling dominates the other.

Based on these theoretical results, we propose and analyze two new leveraging algorithms designed to improve upon vanilla leveraging and uniform sampling algorithms in terms of bias and variance. The first of these (denoted SLEV below) involves increasing the probability of low-leverage samples, and thus it also has the effect of "shrinking" the effect of large leverage scores. The second of these (denoted LEVUNW below) constructs an unweighted version of the leverage-subsampled problem; and thus for a given data set it involves solving a biased subproblem. In both cases, we obtain the algorithmic benefits of leverage-based sampling, while achieving improved statistical performance.

Our main empirical contribution is to provide a detailed evaluation of the statistical properties of these algorithmic leveraging estimators on both synthetic and real data sets. These empirical results indicate that our theory is a good predictor of practical performance for both existing algorithms and our two new leveraging algorithms as well as that our two new algorithms lead to improved performance. In addition, we show that using shrinkage leverage scores typically leads to improved conditional and unconditional biases and variances; and that solving a biased subproblem typically yields improved unconditional biases and variances. By using a recently-developed algorithm of Drineas et al. (2012) to compute fast approximations to the statistical leverage scores, we also demonstrate a regime for large data where our shrinkage leveraging procedure is better algorithmically, in the sense

of computing an answer more quickly than the usual black-box least-squares solver, as well as statistically, in the sense of having smaller mean squared error than naïve uniform sampling. Depending on whether one is interested in results unconditional on the data (which is more traditional from a statistical perspective) or conditional on the data (which is more natural from an algorithmic perspective), we recommend the use of SLEV or LEVUNW, respectively, in the future.

The remainder of this article is organized as follows. We will start in Section 2 with a brief review of linear models, the algorithmic leveraging approach, and related work. Then, in Section 3, we will present our main theoretical results for bias and variance of leveraging estimators. This will be followed in Sections 4 and 5 by a detailed empirical evaluation on a wide range of synthetic and several real data sets. Then, in Section 6, we will conclude with a brief discussion of our results in a broader context. Appendix A will describe our results from the perspective of asymptotic relative efficiency and will consider several toy data sets that illustrate various aspects of algorithmic leveraging; and Appendix B will provide the proofs of our main theoretical results.

2. Background, Notation, and Related Work

In this section, we will provide a brief review of relevant background, including our notation for linear models, an overview of the algorithmic leveraging approach, and a review of related work in statistics and computer science.

2.1 Least-squares and Linear Models

We start with relevant background and notation. Given an $n \times p$ matrix X and an n-dimensional vector $\boldsymbol{y}$, the least-squares (LS) problem is to solve

$$\text{argmin}_{\beta \in \mathbb{R}^p} ||\boldsymbol{y} - X\boldsymbol{\beta}||^2, \tag{1}$$

where $|| \cdot ||$ represents the Euclidean norm on $\mathbb{R}^n$. Of interest is both a vector exactly or approximately optimizing Problem (1), as well as the value of the objective function at the optimum. Using one of several related methods (Golub and Loan, 1996), this LS problem can be solved exactly in $O(np^2)$ time (but, as we will discuss in Section 2.2, it can be solved approximately in $o(np^2)$ time[2]). For example, LS can be solved using the singular value decomposition (SVD): the so-called *thin SVD* of X can be written as $X = U\Lambda V^T$, where U is an $n \times p$ orthogonal matrix whose columns contain the left singular vectors of X, V is an $p \times p$ orthogonal matrix whose columns contain the right singular vectors of X, and the $p \times p$ matrix $\Lambda = Diag\{\lambda_i\}$, where λ_i, $i = 1, \ldots, p$, are the singular values of X. In this case, $\hat{\boldsymbol{\beta}}_{ols} = V\Lambda^{-1}U^T\boldsymbol{y}$.

We consider the use of LS for parameter estimation in a Gaussian linear regression model. Consider the model

$$\boldsymbol{y} = X\boldsymbol{\beta}_0 + \boldsymbol{\epsilon}, \tag{2}$$

2. Recall that, formally, $f(n) = o(g(n))$ as $n \to \infty$ means that for every positive constant ϵ there exists a constant N such that $|f(n)| \leq \epsilon|g(n)|$, for all $n \geq N$. Informally, this means that $f(n)$ grows more slowly than $g(n)$. Thus, if the running time of an algorithm is $o(np^2)$ time, then it is asymptotically faster than any (arbitrarily small) constant times np^2.

where $\boldsymbol{y}$ is an $n \times 1$ response vector, X is an $n \times p$ *fixed* predictor or design matrix, $\boldsymbol{\beta}_0$ is a $p \times 1$ coefficient vector, and the noise vector $\boldsymbol{\epsilon} \sim N(0, \sigma^2 I)$. In this case, the unknown coefficient $\boldsymbol{\beta}_0$ can be estimated via maximum-likelihood estimation as

$$\hat{\boldsymbol{\beta}}_{ols} = \text{argmin}_{\boldsymbol{\beta}} ||\boldsymbol{y} - X\boldsymbol{\beta}||^2 = (X^T X)^{-1} X^T \boldsymbol{y}, \tag{3}$$

in which case the predicted response vector is $\hat{\boldsymbol{y}} = H\boldsymbol{y}$, where $H = X(X^T X)^{-1} X^T$ is the so-called Hat Matrix, which is of interest in classical regression diagnostics (Hoaglin and Welsch, 1978; Chatterjee and Hadi, 1986; Velleman and Welsch, 1981). The i^{th} diagonal element of H, $h_{ii} = \boldsymbol{x}_i^T (X^T X)^{-1} \boldsymbol{x}_i$, where $\boldsymbol{x}_i^T$ is the i^{th} row of X, is the *statistical leverage* of i^{th} observation or sample. The statistical leverage scores have been used historically to quantify the potential of which an observation is an influential observation (Hoaglin and Welsch, 1978; Chatterjee and Hadi, 1986; Velleman and Welsch, 1981), and they will be important for our main results below. Since H can alternatively be expressed as $H = UU^T$, where U is any orthogonal basis for the column space of X, e.g., the Q matrix from a QR decomposition or the matrix of left singular vectors from the thin SVD, the leverage of the i^{th} observation can also be expressed as

$$h_{ii} = \sum_{j=1}^{p} U_{ij}^2 = ||\boldsymbol{u}_i||^2, \tag{4}$$

where $\boldsymbol{u}_i^T$ is the i^{th} row of U. Using Eqn. (4), the exact computation of h_{ii}, for $i \in [n]$, requires $O(np^2)$ time (Golub and Loan, 1996) (but, as we will discuss in Section 2.2, they can be approximated in $o(np^2)$ time).

For an estimate $\hat{\boldsymbol{\beta}}$ of $\boldsymbol{\beta}$, the MSE (mean squared error) associated with the prediction error is defined to be

$$\begin{aligned} MSE(\hat{\boldsymbol{\beta}}) &= \frac{1}{n} \mathbf{E}\left[(X\boldsymbol{\beta}_0 - X\hat{\boldsymbol{\beta}})^T (X\boldsymbol{\beta}_0 - X\hat{\boldsymbol{\beta}}) \right] \\ &= \frac{1}{n} \mathbf{Tr}\left(\mathbf{Var}\left[X\hat{\boldsymbol{\beta}} \right] \right) + \frac{1}{n} (\mathbf{E}\left[X\hat{\boldsymbol{\beta}} \right] - X\boldsymbol{\beta}_0)^T (\mathbf{E}\left[X\hat{\boldsymbol{\beta}} \right] - X\boldsymbol{\beta}_0)) \\ &= \frac{1}{n} \mathbf{Tr}\left(\mathbf{Var}\left[X\hat{\boldsymbol{\beta}} \right] \right) + \frac{1}{n} [\mathbf{bias}(X\hat{\boldsymbol{\beta}})]^T [\mathbf{bias}(X\hat{\boldsymbol{\beta}})] \end{aligned} \tag{5}$$

where $\boldsymbol{\beta}_0$ is the true value of $\boldsymbol{\beta}$. The MSE provides a benchmark to compare the different subsampling estimators, and we will be interested in both the bias and variance components.

2.2 Algorithmic Leveraging for Least-squares Approximation

Here, we will review relevant work on random sampling algorithms for computing approximate solutions to the general overconstrained LS problem (Drineas et al., 2006; Mahoney, 2011; Drineas et al., 2012). These algorithms choose (in general, non-uniformly) a subsample of the data, e.g., a small number of rows of X and the corresponding elements of $\boldsymbol{y}$, and then they perform (typically weighted) LS on the subsample. Importantly, these algorithms make no assumptions on the input data X and $\boldsymbol{y}$, except that $n \gg p$.

A prototypical example of this approach is given by the following meta-algorithm (Drineas et al., 2006; Mahoney, 2011; Drineas et al., 2012), which we call `SubsampleLS`, and which

takes as input an $n \times p$ matrix X, where $n \gg p$, a vector $\boldsymbol{y}$, and a probability distribution $\{\pi_i\}_{i=1}^n$, and which returns as output an approximate solution $\tilde{\boldsymbol{\beta}}_{ols}$, which is an estimate of $\hat{\boldsymbol{\beta}}_{ols}$ of Eqn. (3).

- Randomly sample $r > p$ constraints, i.e., rows of X and the corresponding elements of $\boldsymbol{y}$, using $\{\pi_i\}_{i=1}^n$ as an importance sampling distribution.
- Rescale each sampled row/element by $1/(r\sqrt{\pi_i})$ to form a weighted LS subproblem.
- Solve the weighted LS subproblem, formally given in Eqn. (6) below, and then return the solution $\tilde{\boldsymbol{\beta}}_{ols}$.

It is convenient to describe `SubsampleLS` in terms of a random "sampling matrix" S_X^T and a random diagonal "rescaling matrix" (or "reweighting matrix") D, in the following manner. If we draw r samples (rows or constraints or data points) with replacement, then define an $r \times n$ sampling matrix, S_X^T, where each of the r rows of S_X^T has one non-zero element indicating which row of X (and element of $\boldsymbol{y}$) is chosen in a given random trial. That is, if the k^{th} data unit (or observation) in the original data set is chosen in the i^{th} random trial, then the i^{th} row of S_X^T equals $\mathbf{e}_k$; and thus S_X^T is a random matrix that describes the process of sampling *with* replacement. As an example of applying this sampling matrix, when the sample size $n = 6$ and the subsample size $r = 3$, then premultiplying by

$$S_X^T = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

represents choosing the second, fourth, and fourth data points or samples. The resulting subsample of r data points can be denoted as $(X^*, \boldsymbol{y}^*)$, where $X^*_{r\times p} = S_X^T X$ and $\boldsymbol{y}^*_{r\times 1} = S_X^T \boldsymbol{y}$. In this case, an $r \times r$ diagonal rescaling matrix D can be defined so that i^{th} diagonal element of D equals $1/\sqrt{r\pi_k}$ if the k^{th} data point is chosen in the i^{th} random trial (meaning, in particular, that every diagonal element of D equals $\sqrt{n/r}$ for uniform sampling). With this notation, `SubsampleLS` constructs and solves the *weighted LS estimator*:

$$\text{argmin}_{\beta\in\mathbb{R}^p} ||DS_X^T\boldsymbol{y} - DS_X^T X\boldsymbol{\beta}||^2. \tag{6}$$

Since `SubsampleLS` samples constraints and not variables, the dimensionality of the vector $\tilde{\boldsymbol{\beta}}_{ols}$ that solves the (still overconstrained, but smaller) weighted LS subproblem is the same as that of the vector $\hat{\boldsymbol{\beta}}_{ols}$ that solves the original LS problem. The former may thus be taken as an approximation of the latter, where, of course, the quality of the approximation depends critically on the choice of $\{\pi_i\}_{i=1}^n$. There are several distributions that have been considered previously (Drineas et al., 2006; Mahoney, 2011; Drineas et al., 2012).

- **Uniform Subsampling.** Let $\pi_i = 1/n$, for all $i \in [n]$, i.e., draw the sample uniformly at random.
- **Leverage-based Subsampling.** Let $\pi_i = h_{ii}/\sum_{i=1}^n h_{ii} = h_{ii}/p$ be the normalized statistical leverage scores of Eqn. (4), i.e., draw the sample according to an importance sampling distribution that is proportional to the statistical leverage scores of the data matrix X.

Although Uniform Subsampling (with or without replacement) is very simple to implement, it is easy to construct examples where it will perform very poorly. In particular, it fails dramatically when it is applied to real world data where non-uniform leverage scores are prevalent (e.g., see below or see Drineas et al. 2006; Mahoney 2011). On the other hand, it has been shown that, for a parameter $\gamma \in (0, 1]$ to be tuned, if

$$\pi_i \geq \gamma \frac{h_{ii}}{p}, \text{ and } r = O(p \log(p)/(\gamma\epsilon)), \tag{7}$$

then the following relative-error bounds hold:

$$||\boldsymbol{y} - X\tilde{\boldsymbol{\beta}}_{ols}||_2 \leq (1+\epsilon)||\boldsymbol{y} - X\hat{\boldsymbol{\beta}}_{ols}||_2 \text{ and} \tag{8}$$

$$||\hat{\boldsymbol{\beta}}_{ols} - \tilde{\boldsymbol{\beta}}_{ols}||_2 \leq \sqrt{\epsilon}\left(\kappa(X)\sqrt{\xi^{-2}-1}\right)||\hat{\boldsymbol{\beta}}_{ols}||_2, \tag{9}$$

where $\kappa(X)$ is the condition number of X and where $\xi = ||UU^T\boldsymbol{y}||_2/||\boldsymbol{y}||_2$ is a parameter defining the amount of the mass of $\boldsymbol{y}$ inside the column space of X (Drineas et al., 2006; Mahoney, 2011; Drineas et al., 2012).

Due to the crucial role of the statistical leverage scores in Eqn. (7), we refer to algorithms of the form of `SubsampleLS` as the *algorithmic leveraging* approach to approximating LS approximation. Several versions of the `SubsampleLS` algorithm are of particular interest to us in this paper. We start with two versions that have been studied in the past.

- **Uniform Sampling Estimator (UNIF)** is the estimator resulting from *uniform subsampling* and *weighted LS estimation*, i.e., where Eqn. (6) is solved, where both the sampling and rescaling/reweighting are done with the uniform sampling probabilities. (Note that when the weights are uniform, then the weighted LS estimator of Eqn. (6) leads to the same solution as same as the unweighted LS estimator of Eqn. (11) below.) This version corresponds to vanilla uniform sampling, and it's solution will be denoted by $\tilde{\boldsymbol{\beta}}_{UNIF}$.

- **Basic Leveraging Estimator (LEV)** is the estimator resulting from *exact leverage-based sampling* and *weighted LS estimation*, i.e., where Eqn. (6) is solved, where both the sampling and rescaling/reweighting are done with the leverage-based sampling probabilities given in Eqn. (7). This is the basic algorithmic leveraging algorithm that was originally proposed in (Drineas et al., 2006), where the exact empirical statistical leverage scores of X were first used to construct the subsample and reweight the subproblem, and it's solution will be denoted by $\tilde{\boldsymbol{\beta}}_{LEV}$.

Motivated by our statistical analysis (to come later in the paper), we will introduce two variants of `SubsampleLS`; since these are new to this paper, we also describe them here.

- **Shrinkage Leveraging Estimator (SLEV)** is the estimator resulting from a *shrinkage leverage-based sampling* and *weighted LS estimation*. By shrinkage leverage-based sampling, we mean that we will sample according to a distribution that is a convex combination of a leverage score distribution and the uniform distribution, thereby obtaining the benefits of each; and the rescaling/reweighting is done according to the same distribution. That is, if π^{Lev} denotes a distribution defined by the normalized

leverage scores and π^{Unif} denotes the uniform distribution, then the sampling and reweighting probabilities for SLEV are of the form

$$\pi_i = \alpha\pi_i^{Lev} + (1-\alpha)\pi_i^{Unif}, \tag{10}$$

where $\alpha \in (0,1)$. Thus, with SLEV, Eqn. (6) is solved, where both the sampling and rescaling/reweighting are done with the probabilities given in Eqn. (10). This estimator will be denoted by $\tilde{\boldsymbol{\beta}}_{SLEV}$, and to our knowledge it has not been explicitly considered previously.

- **Unweighted Leveraging Estimator (LEVUNW)** is the estimator resulting from a *leverage-based sampling* and *unweighted LS estimation*. That is, after the samples have been selected with leverage-based sampling probabilities, rather than solving the weighted LS estimator of (6), we will compute the solution of the *unweighted LS estimator*:

$$\text{argmin}_{\beta\in\mathbb{R}^p}||S_X^T\boldsymbol{y} - S_X^T X\boldsymbol{\beta}||^2. \tag{11}$$

  Whereas the previous estimators all follow the basic framework of sampling and rescaling/reweighting according to the same distribution (which is used in worst-case analysis to control the properties of both eigenvalues and eigenvectors and provide unbiased estimates of certain quantities within the analysis, see Drineas et al., 2006; Mahoney, 2011; Drineas et al., 2012), with LEVUNW they are essentially done according to two different distributions—the reason being that not rescaling leads to the same solution as rescaling with the uniform distribution. This estimator will be denoted by $\tilde{\boldsymbol{\beta}}_{LEVUNW}$, and to our knowledge it has not been considered previously.

These methods can all be used to estimate the coefficient vector $\boldsymbol{\beta}$, and we will analyze—both theoretically and empirically—their statistical properties in terms of bias and variance.

2.3 Running Time Considerations

Although it is not our main focus, the running time for leverage-based sampling algorithms is of interest. The running times of these algorithms depend on both the time to construct the probability distribution, $\{\pi_i\}_{i=1}^n$, and the time to solve the subsampled problem. For UNIF, the former is trivial and the latter depends on the size of the subproblem. For estimators that depend on the exact or approximate (recall the flexibility in Eqn. (7) provided by γ) leverage scores, the running time is dominated by the exact or approximate computation of those scores. A naïve algorithm involves using a QR decomposition or the thin SVD of X to obtain the exact leverage scores. Unfortunately, this exact algorithm takes $O(np^2)$ time and is thus no faster than solving the original LS problem exactly. Of greater interest is the algorithm of Drineas et al. (2012) that computes relative-error approximations to all of the leverage scores of X in $o(np^2)$ time.

In more detail, given as input an arbitrary $n \times p$ matrix X, with $n \gg p$, and an error parameter $\epsilon \in (0,1)$, the main algorithm of Drineas et al. (2012) (described also in Section 5.2 below) computes numbers $\tilde{\ell}_i$, for all $i = 1,\ldots,n$, that are relative-error approximations to the leverage scores h_{ii}, in the sense that $|h_{ii} - \tilde{\ell}_i| \leq \epsilon h_{ii}$, for all $i = 1,\ldots,n$. This

algorithm runs in roughly $O(np\log(p)/\epsilon)$ time,[3] which for appropriate parameter settings is $o(np^2)$ time (Drineas et al., 2012). Given the numbers $\tilde{\ell}_i$, for all $i = 1, \ldots, n$, we can let $\pi_i = \tilde{\ell}_i / \sum_{i=1}^n \tilde{\ell}_i$, which then yields probabilities of the form of Eqn. (7) with (say) $\gamma = 0.5$ or $\gamma = 0.9$. Thus, we can use these π_i in place of h_{ii} in BELV, SLEV, or LEVUNW, thus providing a way to implement these procedures in $o(np^2)$ time.

The running time of the relative-error approximation algorithm of Drineas et al. (2012) depends on the time needed to premultiply X by a randomized Hadamard transform (i.e., a "structured" random projection). Recently, high-quality numerical implementations of such random projections have been provided; see, e.g., Blendenpik (Avron et al., 2010), as well as LSRN (Meng et al., 2014), which extends these implementations to large-scale parallel environments. These implementations demonstrate that, for matrices as small as several thousand by several hundred, leverage-based algorithms such as LEV and SLEV can be better in terms of running time than the computation of QR decompositions or the SVD with, e.g., Lapack. See Avron et al. (2010); Meng et al. (2014) for details, and see Gittens and Mahoney (2013) for the application of these methods to the fast computation of leverage scores. Below, we will evaluate an implementation of a variant of the main algorithm of Drineas et al. (2012) in the software environment R.

2.4 Additional Related Work

Our leverage-based methods for estimating $\boldsymbol{\beta}$ are related to resampling methods such as the bootstrap (Efron, 1979), and many of these resampling methods enjoy desirable asymptotic properties (Shao and Tu, 1995). Resampling methods in linear models were studied extensively in Wu (1986) and are related to the jackknife (Miller, 1974a,b; Jaeckel, 1972; Efron and Gong, 1983). They usually produce resamples at a similar size to that of the full data, whereas algorithmic leveraging is primarily interested in constructing subproblems that are much smaller than the full data. In addition, the goal of resampling is traditionally to perform statistical inference and not to improve the running time of an algorithm, except in the very recent work (Kleiner et al., 2012). Additional related work in statistics includes Hinkley (1977); Rubin (1981); Liu et al. (1998); Bickel et al. (1997); Politis et al. (1999).

After the submission to JMLR, we were made aware, by the reviewers, of two related pieces of work (Dhillon et al., 2013; Hsu et al., 2014). Dhillon et al. (2013) analyzed the random rotation and uniform sampling, and then proposed several sampling procedures that were justified in a statistical setting. For these sampling procedures, Dhillon et al. (2013) derived some error bounds, which are in the same line of thinking as Drineas et al. (2006, 2010). Hsu et al. (2014) applied a uniform sampling analysis to matrix X after random rotation and derived prediction error bound.

3. Bias and Variance Analysis of Subsampling Estimators

In this section, we develop analytic methods to study the biases and variances of the subsampling estimators described in Section 2.2. Analyzing these subsampling methods is

3. In more detail, the asymptotic running time of the main algorithm of Drineas et al. (2012) is $O\left(np\ln\left(p\epsilon^{-1}\right) + np\epsilon^{-2}\ln n + p^3\epsilon^{-2}\left(\ln n\right)\left(\ln\left(p\epsilon^{-1}\right)\right)\right)$. To simplify this expression, suppose that $p \leq n \leq e^p$ and treat ϵ as a constant; then, the asymptotic running time is $O\left(np\ln n + p^3\left(\ln n\right)\left(\ln p\right)\right)$.

challenging for at least the following two reasons: first, there are two layers of randomness in the estimators, i.e., the randomness inherent in the linear regression model as well as random subsampling of a particular sample from the linear model; and second, the estimators depends on random subsampling through the inverse of random sampling matrix, which is a nonlinear function. To ease the analysis, we will employ a Taylor series analysis to approximate the subsampling estimators as linear combinations of random sampling matrices, and we will consider biases and variances both conditioned as well as not conditioned on the data. Here is a brief outline of the main results of this section.

- We will start in Section 3.1 with bias and variance results for weighted LS estimators for general sampling/reweighting probabilities. This will involve viewing the solution of the subsampled LS problem as a function of the vector of sampling/reweighting probabilities and performing a Taylor series expansion of the solution to the subsampled LS problem around the expected value (where the expectation is taken with respect to the random choices of the algorithm) of that vector.

- Then, in Section 3.2, we will specialize these results to leverage-based sampling and uniform sampling, describing their complementary properties. We will see that, in terms of bias and variance, neither LEV nor UNIF is uniformly better than the other. In particular, LEV has variance whose size-scale is better than the size-scale of UNIF; but UNIF does not have leverage scores in the denominator of its variance expressions, as does LEV, and thus the variance of UNIF is not inflated on inputs that have very small leverage scores.

- Finally, in Section 3.3, we will propose and analyze two new leveraging algorithms that will address deficiencies of LEV and UNIF in two different ways. The first, SLEV, constructs a smaller LS problem with "shrinkage" leverage scores that are constructed as a convex combination of leverage score probabilities and uniform probabilities; and the second, LEVUNW, uses leverage-based sampling probabilities to construct and solve an unweighted or biased LS problem.

3.1 Traditional Weighted Sampling Estimators

We start with the bias and variance of the traditional weighted sampling estimator $\tilde{\boldsymbol{\beta}}_W$, given in Eqn. (12) below. Recall that this estimator actually refers to a parameterized family of estimators, parameterized by the sampling/rescaling probabilities. The estimate obtained by solving the weighted LS problem of (6) can be represented as

$$\begin{aligned} \tilde{\boldsymbol{\beta}}_W &= (X^T S_X D^2 S_X^T X)^{-1} X^T S_X^T D^2 S_X \boldsymbol{y} \\ &= (X^T W X)^{-1} X^T W \boldsymbol{y}, \end{aligned} \tag{12}$$

where $W = S_X D^2 S_X^T$ is an $n \times n$ diagonal random matrix, i.e., all off-diagonal elements are zeros, and where both S_X and D are defined in terms of the sampling/rescaling probabilities. (In particular, W describes the probability distribution with which to draw the sample *and* with which to reweigh the subsample, where both are done according to the same distribution. Thus, this section does *not* apply to LEVUNW; see Section 3.3.2 for the extension to LEVUNW.) Although our results hold more generally, we are most interested in UNIF, LEV, and SLEV, as described in Section 2.2.

Clearly, the vector $\tilde{\boldsymbol{\beta}}_W$ can be regarded as a function of the random weight vector $\boldsymbol{w} = (w_1, w_2, \ldots, w_n)^T$, denoted as $\tilde{\boldsymbol{\beta}}_W(\boldsymbol{w})$, where $(w_1, w_2, \ldots, w_n)$ are diagonal entries of W. Since we are performing random sampling with replacement, it is easy to see that $\boldsymbol{w} = (w_1, w_2, \ldots, w_n)^T$ has a scaled multinomial distribution,

$$\mathbf{Pr}\left[w_1 = \frac{k_1}{r\pi_1}, w_2 = \frac{k_2}{r\pi_2}, \ldots, w_n = \frac{k_n}{r\pi_n}\right] = \frac{r!}{k_1!k_2!\ldots,k_n!}\pi_1^{k_1}\pi_2^{k_2}\cdots\pi_n^{k_n},$$

and thus it can easily be shown that $\mathbf{E}[\boldsymbol{w}] = \mathbf{1}$. By setting $\boldsymbol{w}_0$, the vector around which we will perform our Taylor series expansion, to be the all-ones vector, i.e., $\boldsymbol{w}_0 = \mathbf{1}$, then $\tilde{\boldsymbol{\beta}}(\boldsymbol{w})$ can be expanded around the full sample ordinary LS estimate $\hat{\boldsymbol{\beta}}_{ols}$, i.e., $\tilde{\boldsymbol{\beta}}_W(\mathbf{1}) = \hat{\boldsymbol{\beta}}_{ols}$. From this, we can establish the following lemma, the proof of which may be found in Appendix B.

Lemma 1 *Let $\tilde{\boldsymbol{\beta}}_W$ be the output of the `SubsampleLS` Algorithm, obtained by solving the weighted LS problem of (6). Then, a Taylor expansion of $\tilde{\boldsymbol{\beta}}_W$ around the point $\boldsymbol{w}_0 = \mathbf{1}$ yields*

$$\tilde{\boldsymbol{\beta}}_W = \hat{\boldsymbol{\beta}}_{ols} + (X^TX)^{-1}X^T Diag\left\{\hat{\boldsymbol{e}}\right\}(\boldsymbol{w} - \mathbf{1}) + R_W, \tag{13}$$

where $\hat{\boldsymbol{e}} = \boldsymbol{y} - X\hat{\boldsymbol{\beta}}_{ols}$ is the LS residual vector, and where R_W is the Taylor expansion remainder.

Remark. The significance of Lemma 1 is that, to leading order, the vector $\boldsymbol{w}$ that encodes information about the sampling process and subproblem construction enters the estimator of $\tilde{\boldsymbol{\beta}}_W$ linearly. The additional error, R_W depends strongly on the details of the sampling process, and in particular will be very different for UNIF, LEV, and SLEV.

Remark. Our approximations hold when the Taylor series expansion is valid, i.e., when R_W is "small," e.g., $R_W = o_p(||\boldsymbol{w} - \boldsymbol{w}_0||)$, where $o_p(\cdot)$ means "little o" with high probability over the randomness in the random vector $\boldsymbol{w}$. Although we will evaluate the quality of our approximations empirically in Sections 4 and 5, we currently do *not* have a precise theoretical characterization of when this holds. Here, we simply make two observations. First, this expression will fail to hold if rank is lost in the sampling process. This is because in general there will be a bias due to failing to capture information in the dimensions that are not represented in the sample (Recall that one may use the Moore-Penrose generalized inverse for inverting rank-deficient matrices). Second, this expression will tend to hold better as the subsample size r is increased. However, for a fixed value of r, the linear approximation regime will be larger when the sample is constructed using information in the leverage scores—since, among other things, using leverage scores in the sampling process is designed to preserve the rank of the subsampled problem (Drineas et al., 2006; Mahoney, 2011; Drineas et al., 2012). A detailed discussion of this last point is available in Mahoney (2011); and these observations will be confirmed empirically in Section 5.

Remark. Since, essentially, LEVUNW involves sampling and reweighting according to two *different* distributions[4], the analogous expression for LEVUNW will be somewhat different, as will be discussed in Lemma 5 in Section 3.3.

4. In this case, the latter distribution is the uniform distribution, where recall that reweighting uniformly leads to the same solution as not reweighting at all.

Given Lemma 1, we can establish the following lemma, which provides expressions for the conditional and unconditional expectations and variances for the weighted sampling estimators. The first two expressions in the lemma are conditioned on the data vector $\boldsymbol{y}$[5]; and the last two expressions in the lemma provide similar results, except that they are not conditioned on the data vector $\boldsymbol{y}$. The proof of this lemma appears in Appendix B.

Lemma 2 *The conditional expectation and conditional variance for the traditional algorithmic leveraging procedure, i.e., when the subproblem solved is a weighted LS problem of the form (6), are given by:*

$$\mathbf{E_w}\left[\tilde{\boldsymbol{\beta}}_W|\boldsymbol{y}\right] = \hat{\boldsymbol{\beta}}_{ols} + \mathbf{E_w}\left[R_W\right]; \tag{14}$$

$$\begin{aligned}\mathbf{Var_w}\left[\tilde{\boldsymbol{\beta}}_W|\boldsymbol{y}\right] = &(X^TX)^{-1}X^T\left[Diag\left\{\hat{\boldsymbol{e}}\right\}Diag\left\{\frac{1}{r\boldsymbol{\pi}}\right\}Diag\left\{\hat{\boldsymbol{e}}\right\}\right]X(X^TX)^{-1}\\ &+\quad \mathbf{Var_w}\left[R_W\right],\end{aligned} \tag{15}$$

where W specifies the probability distribution used in the sampling and rescaling steps. The unconditional expectation and unconditional variance for the traditional algorithmic leveraging procedure are given by:

$$\mathbf{E}\left[\tilde{\boldsymbol{\beta}}_W\right] = \boldsymbol{\beta}_0; \tag{16}$$

$$\begin{aligned}\mathbf{Var}\left[\tilde{\boldsymbol{\beta}}_W\right] = &\sigma^2(X^TX)^{-1} + \frac{\sigma^2}{r}(X^TX)^{-1}X^TDiag\left\{\frac{(1-h_{ii})^2}{\pi_i}\right\}X(X^TX)^{-1}\\ &+\quad \mathbf{Var}\left[R_W\right].\end{aligned} \tag{17}$$

Remark. Eqn. (14) states that, when the $\mathbf{E_w}\left[R_W\right]$ term is negligible, i.e., when the linear approximation is valid, then, conditioning on the observed data $\boldsymbol{y}$, the estimate $\tilde{\boldsymbol{\beta}}_W$ is approximately unbiased, relative to the full sample ordinarily LS estimate $\hat{\boldsymbol{\beta}}_{ols}$; and Eqn. (16) states that the estimate $\tilde{\boldsymbol{\beta}}_W$ is unbiased, relative to the "true" value $\boldsymbol{\beta}_0$ of the parameter vector $\boldsymbol{\beta}$. That is, given a particular data set $(X, \boldsymbol{y})$, the conditional expectation result of Eqn. (14) states that the leveraging estimators can approximate well $\hat{\boldsymbol{\beta}}_{ols}$; and, as a statistical inference procedure for arbitrary data sets, the unconditional expectation result of Eqn. (16) states that the leveraging estimators can infer well $\boldsymbol{\beta}_0$.

Remark. Both the conditional variance of Eqn. (15) and the (second term of the) unconditional variance of Eqn. (17) are inversely proportional to the subsample size r; and both contain a sandwich-type expression, the middle of which depends on how the leverage scores interact with the sampling probabilities. Moreover, the first term of the unconditional variance, $\sigma^2(X^TX)^{-1}$, equals the variance of the ordinary LS estimator; this implies, e.g., that the unconditional variance of Eqn. (17) is larger than the variance of the ordinary LS estimator, which is consistent with the Gauss-Markov theorem.

3.2 Leverage-based Sampling and Uniform Sampling Estimators

Here, we specialize Lemma 2 by stating two lemmas that provide the conditional and unconditional expectation and variance for LEV and UNIF, and we will discuss the relative

5. Here and below, the subscript $\mathbf{w}$ on $\mathbf{E_w}$ and $\mathbf{Var_w}$ refers to performing expectations and variances with respect to (just) the random weight vector $\boldsymbol{w}$ and not the data.

merits of each procedure. The proofs of these two lemmas are immediate, given the proof of Lemma 2. Thus, we omit the proofs, and instead discuss properties of the expressions that are of interest in our empirical evaluation.

Our main conclusion here is that Lemma 3 and Lemma 4 highlight that the statistical properties of the algorithmic leveraging method can be quite different than the algorithmic properties. Prior work has adopted an *algorithmic perspective* that has focused on providing worst-case running time bounds for arbitrary input matrices. From this algorithmic perspective, leverage-based sampling (i.e., explicitly or implicitly biasing toward high-leverage components, as is done in particular with the LEV procedure) provides uniformly superior worst-case algorithmic results, when compared with UNIF (Drineas et al., 2006; Mahoney, 2011; Drineas et al., 2012). Our analysis here reveals that, from a *statistical perspective* where one is interested in the bias and variance properties of the estimators, the situation is considerably more subtle. In particular, a key conclusion from Lemmas 3 and 4 is that, with respect to their variance or MSE, neither LEV nor UNIF is uniformly superior for all input.

We start with the bias and variance of the leverage subsampling estimator $\tilde{\boldsymbol{\beta}}_{LEV}$.

Lemma 3 *The conditional expectation and conditional variance for the LEV procedure are given by:*

$$\begin{aligned}
\mathbf{E}_{\mathbf{w}}\left[\tilde{\boldsymbol{\beta}}_{LEV}|\boldsymbol{y}\right] &= \hat{\boldsymbol{\beta}}_{ols} + \mathbf{E}_{\mathbf{w}}\left[R_{LEV}\right]; \\
\mathbf{Var}_{\mathbf{w}}\left[\tilde{\boldsymbol{\beta}}_{LEV}|\boldsymbol{y}\right] &= \frac{p}{r}(X^TX)^{-1}X^T\left[Diag\left\{\hat{\boldsymbol{e}}\right\}Diag\left\{\frac{1}{h_{ii}}\right\}Diag\left\{\hat{\boldsymbol{e}}\right\}\right]X(X^TX)^{-1} \\
&\quad + \quad \mathbf{Var}_{\mathbf{w}}\left[R_{LEV}\right].
\end{aligned}$$

The unconditional expectation and unconditional variance for the LEV procedure are given by:

$$\begin{aligned}
\mathbf{E}\left[\tilde{\boldsymbol{\beta}}_{LEV}\right] &= \boldsymbol{\beta}_0; \\
\mathbf{Var}\left[\tilde{\boldsymbol{\beta}}_{LEV}\right] &= \sigma^2(X^TX)^{-1} + \frac{p\sigma^2}{r}(X^TX)^{-1}X^TDiag\left\{\frac{(1-h_{ii})^2}{h_{ii}}\right\}X(X^TX)^{-1} \\
&\quad + \quad \mathbf{Var}\left[R_{LEV}\right].
\end{aligned} \tag{18}$$

Remark. Two points are worth making. First, the variance expressions for LEV depend on the size (i.e., the number of columns and rows) of the $n \times p$ matrix X and the number of samples r as p/r. This variance size-scale many be made to be very small if $p \ll r \ll n$. Second, the sandwich-type expression depends on the leverage scores as $1/h_{ii}$, implying that the variances could be inflated to arbitrarily large values by very small leverage scores. Both of these observations will be confirmed empirically in Section 4.

We next turn to the bias and variance of the uniform subsampling estimator $\tilde{\boldsymbol{\beta}}_{UNIF}$.

Lemma 4 *The conditional expectation and conditional variance for the UNIF procedure are given by:*

$$\begin{aligned}\mathbf{E_w}\left[\tilde{\boldsymbol{\beta}}_{UNIF}|\boldsymbol{y}\right] &= \hat{\boldsymbol{\beta}}_{ols} + \mathbf{E_w}\left[R_{UNIF}\right] \\ \mathbf{Var_w}\left[\tilde{\boldsymbol{\beta}}_{UNIF}|\boldsymbol{y}\right] &= \frac{n}{r}(X^TX)^{-1}X^T\left[Diag\left\{\hat{\boldsymbol{e}}\right\}Diag\left\{\hat{\boldsymbol{e}}\right\}\right]X(X^TX)^{-1} \\ &+ \quad \mathbf{Var_w}\left[R_{UNIF}\right]. \end{aligned} \tag{19}$$

The unconditional expectation and unconditional variance for the UNIF procedure are given by:

$$\begin{aligned}\mathbf{E}\left[\tilde{\boldsymbol{\beta}}_{UNIF}\right] &= \boldsymbol{\beta}_0; \\ \mathbf{Var}\left[\tilde{\boldsymbol{\beta}}_{UNIF}\right] &= \sigma^2(X^TX)^{-1} + \frac{n}{r}\sigma^2(X^TX)^{-1}X^TDiag\left\{(1-h_{ii})^2\right\}X(X^TX)^{-1} \\ &+ \quad \mathbf{Var}\left[R_{UNIF}\right]. \end{aligned} \tag{20}$$

Remark. Two points are worth making. First, the variance expressions for UNIF depend on the size (i.e., the number of columns and rows) of the $n \times p$ matrix X and the number of samples r as n/r. Since this variance size-scale is very large, e.g., compared to the p/r from LEV, these variance expressions will be large unless r is nearly equal to n. Second, the sandwich-type expression is not inflated by very small leverage scores.
Remark. Apart from a factor n/r, the conditional variance for UNIF, as given in Eqn. (19), is the same as Hinkley's weighted jackknife variance estimator (Hinkley, 1977).

3.3 Novel Leveraging Estimators

In view of Lemmas 3 and 4, we consider several ways to take advantage of the complementary strengths of the LEV and UNIF procedures. Recall that we would like to sample with respect to probabilities that are "near" those defined by the empirical statistical leverage scores. We at least want to identify large leverage scores to preserve rank. This helps ensure that the linear regime of the Taylor expansion is large, and it also helps ensure that the scale of the variance is p/r and not n/r. But we would like to avoid rescaling by $1/h_{ii}$ when certain leverage scores are extremely small, thereby avoiding inflated variance estimates.

3.3.1 The Shrinkage Leveraging (SLEV) Estimator

Consider first the SLEV procedure. As described in Section 2.2, this involves sampling and reweighting with respect to a distribution that is a convex combination of the empirical leverage score distribution and the uniform distribution. That is, let π^{Lev} denote a distribution defined by the normalized leverage scores (i.e., $\pi_i^{Lev} = h_{ii}/p$, or π^{Lev} is constructed from the output of the algorithm of (Drineas et al., 2012) that computes relative-error approximations to the leverage scores), and let π^{Unif} denote the uniform distribution (i.e., $\pi_i^{Unif} = 1/n$, for all $i \in [n]$); then the sampling probabilities for the SLEV procedure are of the form

$$\pi_i = \alpha\pi_i^{Lev} + (1-\alpha)\pi_i^{Unif}, \tag{21}$$

where $\alpha \in (0,1)$.

Since SLEV involves solving a weighted LS problem of the form of Eqn. (6), expressions of the form provided by Lemma 2 hold immediately. In particular, SLEV enjoys approximate unbiasedness, in the same sense that the LEV and UNIF procedures do. The particular expressions for the higher order terms can be easily derived, but they are much messier and less transparent than the bounds provided by Lemmas 3 and 4 for LEV and UNIF, respectively. Thus, rather than presenting them, we simply point out several aspects of the SLEV procedure that should be immediate, given our earlier theoretical discussion.

First, note that $\min_i \pi_i \geq (1-\alpha)/n$, with equality obtained when $h_{ii} = 0$. Thus, assuming that $1-\alpha$ is not extremely small, e.g., $1-\alpha = 0.1$, then none of the SLEV sampling probabilities is too small, and thus the variance of the SLEV estimator does not get inflated too much, as it could with the LEV estimator. Second, assuming that $1-\alpha$ is not too large, e.g., $1-\alpha = 0.1$, then Eqn. (7) is satisfied with $\gamma = 1.1$, and thus the amount of oversampling that is required, relative to the LEV procedure, is not much, e.g., 10%. In this case, the variance of the SLEV procedure has a scale of p/r, as opposed to n/r scale of UNIF, assuming that r is increased by that 10%. Third, since Eqn. (21) is still required to be a probability distribution, combining the leverage score distribution with the uniform distribution has the effect of not only increasing the very small scores, but it also has the effect of performing shrinkage on the very large scores. Finally, all of these observations also hold if, rather that using the exact leverage score distribution (which recall takes $O(np^2)$ time to compute), we instead use approximate leverage scores, as computed with the fast algorithm of Drineas et al. (2012). For this reason, this approximate version of the SLEV procedure is the most promising for very large-scale applications.

3.3.2 The Unweighted Leveraging (LEVUNW) Estimator

Consider next the LEVUNW procedure. As described in Section 2.2, this estimator is different than the previous estimators, in that the sampling and reweighting are done according to different distributions. (Since LEVUNW does *not* sample and reweight according to the same probability distribution, our previous analysis does not apply.) Thus, we shall examine the bias and variance of the unweighted leveraging estimator $\tilde{\boldsymbol{\beta}}_{LEVUNW}$. To do so, we first use a Taylor series expansion to get the following lemma, the proof of which may be found in Appendix B.

Lemma 5 *Let $\tilde{\boldsymbol{\beta}}_{LEVUNW}$ be the output of the modified `SubsampleLS` Algorithm, obtained by solving the unweighted LS problem of (11). Then, a Taylor expansion of $\tilde{\boldsymbol{\beta}}_{LEVUNW}$ around the point $\boldsymbol{w}_0 = r\boldsymbol{\pi}$ yields*

$$\tilde{\boldsymbol{\beta}}_{LEVUNW} = \hat{\boldsymbol{\beta}}_{wls} + (X^T W_0 X)^{-1} X^T Diag\left\{\hat{\boldsymbol{e}}_w\right\} (\boldsymbol{w} - r\boldsymbol{\pi}) + R_{LEVUNW}, \tag{22}$$

where $\hat{\boldsymbol{\beta}}_{wls} = (X^T W_0 X)^{-1} X W_0 \boldsymbol{y}$ is the full sample weighted LS estimator, $\hat{\boldsymbol{e}}_w = \boldsymbol{y} - X\hat{\boldsymbol{\beta}}_{wls}$ is the LS residual vector, $W_0 = Diag\left\{r\boldsymbol{\pi}\right\} = Diag\left\{rh_{ii}/p\right\}$, and R_{LEVUNW} is the Taylor expansion remainder.

Remark. This lemma is analogous to Lemma 1. Since the sampling and reweighting are performed according to different distributions, however, the point about which the Taylor expansion is performed, as well as the prefactors of the linear term, are somewhat different.

In particular, here we expand around the point $\boldsymbol{w}_0 = r\boldsymbol{\pi}$ since $\mathbf{E}[\boldsymbol{w}] = r\boldsymbol{\pi}$ when no reweighting takes place.

Given this Taylor expansion lemma, we can now establish the following lemma for the mean and variance of LEVUNW, both conditioned and unconditioned on the data $\boldsymbol{y}$. The proof of the following lemma may be found in Appendix B.

Lemma 6 *The conditional expectation and conditional variance for the LEVUNW procedure are given by:*

$$\begin{aligned}
\mathbf{E_w}\left[\tilde{\boldsymbol{\beta}}_{LEVUNW}|\boldsymbol{y}\right] &= \hat{\boldsymbol{\beta}}_{wls} + \mathbf{E_w}\left[R_{LEVUNW}\right]; \\
\mathbf{Var_w}\left[\tilde{\boldsymbol{\beta}}_{LEVUNW}|\boldsymbol{y}\right] &= (X^TW_0X)^{-1}X^T Diag\left\{\hat{\boldsymbol{e}}_w\right\} W_0 Diag\left\{\hat{\boldsymbol{e}}_w\right\} X(X^TW_0X)^{-1} \\
&+ \mathbf{Var_w}\left[R_{LEVUNW}\right],
\end{aligned}$$

where $W_0 = Diag\{r\boldsymbol{\pi}\}$*, and where* $\hat{\boldsymbol{\beta}}_{wls} = (X^TW_0X)^{-1}XW_0\boldsymbol{y}$ *is the full sample weighted LS estimator. The unconditional expectation and unconditional variance for the LEVUNW procedure are given by:*

$$\begin{aligned}
\mathbf{E}\left[\tilde{\boldsymbol{\beta}}_{LEVUNW}\right] &= \boldsymbol{\beta}_0; \\
\mathbf{Var}\left[\tilde{\boldsymbol{\beta}}_{LEVUNW}\right] &= \sigma^2(X^TW_0X)^{-1}X^TW_0^2X(X^TW_0X)^{-1} \\
&+ \sigma^2(X^TW_0X)^{-1}X^T Diag\left\{I - P_{X,W_0}\right\} W_0 Diag\left\{I - P_{X,W_0}\right\} X \\
&\quad (X^TW_0X)^{-1} + \mathbf{Var}\left[R_{LEVUNW}\right] \qquad (23)
\end{aligned}$$

where $P_{X,W_0} = X(X^TW_0X)^{-1}X^TW_0$.

Remark. The two expectation results in this lemma state: (i), when $\mathbf{E_w}[R_{LEVUNW}]$ is negligible, then, conditioning on the observed data $\boldsymbol{y}$, the estimator $\tilde{\boldsymbol{\beta}}_{LEVUNW}$ is approximately unbiased, relative to the full sample *weighted* LS estimator $\hat{\boldsymbol{\beta}}_{wls}$; and (ii) the estimator $\tilde{\boldsymbol{\beta}}_{LEVUNW}$ is unbiased, relative to the "true" value $\boldsymbol{\beta}_0$ of the parameter vector $\boldsymbol{\beta}$. That is, if we apply LEVUNW to a given data set N times, then the average of the N LEVUNW estimates are *not* centered at the LS estimate, but instead are centered roughly at the weighted least squares estimate; while if we generate many data sets from the true model and apply LEVUNW to these data sets, then the average of these estimates is centered around true value $\boldsymbol{\beta}_0$.

Remark. As expected, when the leverage scores are all the same, the variance in Eqn. (23) is the same as the variance of uniform random sampling. This is expected since, when reweighting with respect to the uniform distribution, one does not change the problem being solved, and thus the solutions to the weighted and unweighted LS problems are identical. More generally, the variance is not inflated by very small leverage scores, as it is with LEV. For example, the conditional variance expression is also a sandwich-type expression, the center of which is $W_0 = Diag\{rh_{ii}/n\}$, which is not inflated by very small leverage scores.

4. Main Empirical Evaluation

In this section, we describe the main part of our empirical analysis of the behavior of the biases and variances of the subsampling estimators described in Section 2.2. Additional

empirical results will be presented in Section 5. In these two sections, we will use both synthetic data and real data to illustrate the extreme properties of the subsampling methods in realistic settings. We will use the MSE as a benchmark to compare the different subsampling estimators; but since we are interested in both the bias and variance properties of our estimates, we will present results for both the bias and variance separately.

Here is a brief outline of the main results of this section.

- In Section 4.1, we will describe our synthetic data. These data are drawn from three standard distributions, and they are designed to provide relatively-realistic synthetic examples where leverage scores are fairly uniform, moderately nonuniform, or very nonuniform.
- Then, in Section 4.2, we will summarize our results for the unconditional bias and variance for LEV and UNIF, when applied to the synthetic data.
- Then, in Section 4.3, we will summarize our results for the unconditional bias and variance of SLEV and LEVUNW. This will illustrate that both SLEV and LEVUNW can overcome some of the problems associated with LEV and UNIF.
- Finally, in Section 4.4, we will present our results for the conditional bias and variance of SLEV and LEVUNW (as well as LEV and UNIF). In particular, this will show that LEVUNW can incur substantial bias, relative to the other methods, when conditioning on a given data set.

4.1 Description of Synthetic Data

We consider synthetic data of 1000 runs generated from $\boldsymbol{y} = X\boldsymbol{\beta} + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim N(0, 9I_n)$, where several different values of n and p, leading to both "very rectangular" and "moderately rectangular" matrices X, are considered. The design matrix X is generated from one of three different classes of distributions introduced below. These three distributions were chosen since the first has nearly uniform leverage scores, the second has mildly non-uniform leverage scores, and the third has very non-uniform leverage scores.

- **Nearly uniform leverage scores (GA).** We generated an $n \times p$ matrix X from multivariate normal $N(\mathbf{1}_p, \Sigma)$, where the (i,j)th element of $\Sigma_{ij} = 2 \times 0.5^{|i-j|}$, and where we set $\boldsymbol{\beta} = (\mathbf{1}_{10}, 0.1\mathbf{1}_{p-20}, \mathbf{1}_{10})^T$. (Referred to as GA data.)
- **Moderately nonuniform leverage scores (T_3).** We generated X from multivariate t-distribution with 3 degree of freedom and covariance matrix Σ as before. (Referred to as T_3 data.)
- **Very nonuniform leverage scores (T_1).** We generated X from multivariate t-distribution with 1 degree of freedom and covariance matrix Σ as before. (Referred to as T_1 data.)

See Table 4.1 for a summary of the parameters for the synthetic data we considered and for basic summary statistics for the leverage scores probabilities (i.e., the leverage scores that have been normalized to sum to 1 by dividing by p) of these data matrices. The results reported in Table 4.1 are for leverage score statistics for a single fixed data matrix

Distn	n	p	Min	Median	Max	Mean	Std.Dev.	$\frac{\text{Max}}{\text{Min}}$	$\frac{\text{Max}}{\text{Median}}$
GA	1K	10	1.96e-4	9.24e-4	2.66e-3	1.00e-3	4.49e-4	13.5	2.88
GA	1K	50	4.79e-4	9.90e-4	1.74e-3	1.00e-3	1.95e-4	3.63	1.76
GA	1K	100	6.65e-4	9.94e-4	1.56e-3	1.00e-3	1.33e-4	2.35	1.57
GA	5K	10	1.45e-5	1.88e-4	6.16e-4	2.00e-4	8.97e-5	42.4	3.28
GA	5K	50	9.02e-5	1.98e-4	3.64e-4	2.00e-4	3.92e-5	4.03	1.84
GA	5K	250	1.39e-4	1.99e-4	2.68e-4	2.00e-4	1.73e-5	1.92	1.34
GA	5K	500	1.54e-4	2.00e-4	2.48e-4	2.00e-4	1.20e-5	1.61	1.24
T_3	1K	10	2.64e-5	4.09e-4	5.63e-2	1.00e-3	2.77e-3	2.13e+3	138
T_3	1K	50	6.57e-5	5.21e-4	1.95e-2	1.00e-3	1.71e-3	297	37.5
T_3	1K	100	7.26e-5	6.39e-4	9.04e-3	1.00e-3	1.06e-3	125	14.1
T_3	5K	10	5.23e-6	7.73e-5	5.85e-2	2.00e-4	9.66e-4	1.12e+4	757
T_3	5K	50	9.60e-6	9.84e-5	1.52e-2	2.00e-4	4.64e-4	1.58e+3	154
T_3	5K	250	1.20e-5	1.14e-4	3.56e-3	2.00e-4	2.77e-4	296	31.2
T_3	5K	500	1.72e-5	1.29e-4	1.87e-3	2.00e-4	2.09e-4	108	14.5
T_1	1K	10	4.91e-8	4.52e-6	9.69e-2	1.00e-3	8.40e-3	1.97e+6	2.14e+4
T_1	1K	50	2.24e-6	6.18e-5	2.00e-2	1.00e-3	3.07e-3	8.93e+3	323
T_1	1K	100	4.81e-6	1.66e-4	9.99e-3	1.00e-3	2.08e-3	2.08e+3	60.1
T_1	5K	10	5.00e-9	6.18e-7	9.00e-2	2.00e-4	3.00e-3	1.80e+7	1.46e+5
T_1	5K	50	4.10e-8	2.71e-6	2.00e-2	2.00e-4	1.39e-3	4.88e+5	7.37e+3
T_1	5K	250	3.28e-7	1.50e-5	4.00e-3	2.00e-4	6.11e-4	1.22e+4	267
T_1	5K	500	1.04e-6	2.79e-5	2.00e-3	2.00e-4	4.24e-4	1.91e+3	71.6

Table 1: Summary statistics for leverage-score probabilities (i.e., leverage scores divided by p) for the synthetic data sets.

X generated in the above manner (for each of the 3 procedures and for each value of n and p), but we have confirmed that similar results hold for other matrices X generated in the same manner.

Several observations are worth making about the summaries presented in Table 4.1. First, and as expected, the Gaussian data tend to have the most uniform leverage scores, the T_3 data are intermediate, and the T_1 data have the most nonuniform leverage scores, as measured by both the standard deviation of the scores as well as the ratio of maximum to minimum leverage score. Second, the standard deviation of the leverage score distribution is substantially less sensitive to non-uniformities in the leverage scores than is the ratio of the maximum to minimum leverage score (or the maximum to the mean/median score, although all four measures exhibit the same qualitative trends). Although we have not pursued it, this suggests that these latter measures will be more informative as to when leverage-based sampling might be necessary in a particular application. Third, in all these cases, the variability trends are caused both by the large (in particular, the maximum)

leverage scores increasing as well as the small (in particular, the minimum) leverage scores decreasing. Fourth, within a given type of distribution (i.e., GA or T_3 or T_1), leverage scores are more nonuniform when the matrix X is more rectangular, and this is true both when n is held fixed and when p is held fixed.

4.2 Leveraging Versus Uniform Sampling on Synthetic Data

Here, we will describe the properties of LEV versus UNIF for synthetic data. See Figures 1, 2, and 3 for the results on data matrices with $n = 1000$ and $p = 10$, 50, and 100, respectively. (The results for data matrices for $n = 5000$ and other values of n are similar.) In each case, we generated a single matrix from that distribution (which we then fixed to generate the $\boldsymbol{y}$ vectors) and $\boldsymbol{\beta}_0$ was set to be the all-ones vector; and then we ran the sampling process multiple times, typically ca. 1000 times, in order to obtain reliable estimates for the biases and variances. In each of the Figures 1, 2, and 3, the top panel is the variance, the bottom panel is the squared bias; for both the bias and variance, we have plotted the results in log-scale; and, in each figure, the first column is the GA model, the middle column is the T_3 model, and the right column is the T_1 model.

The simulation results corroborate what we have learned from our theoretical analysis, and there are several things worth noting. First, in general the squared bias is much less than the variance, even for the T_1 data, suggesting that the solution is unbiased in the sense quantified in Lemmas 3 and 4. Second, LEV and UNIF perform very similarly for GA, somewhat less similarly for T_3, and quite differently for T_1, consistent with the results in Table 4.1 indicating that the leverage scores are very uniform for GA and very nonuniform for T_1. In addition, when they are different, LEV tends to perform better than UNIF, i.e., have a lower MSE for a fixed sampling complexity. Third, as the subsample size increases, the squared bias and variance tend to decrease monotonically. In particular, the variance tends to decrease roughly as $1/r$, where r is the size of the subsample, in agreement with Lemmas 3 and 4. Moreover, the decrease for UNIF is much slower, in a manner more consistent with the leading term of n/r in Eqn. (20), than is the decrease for LEV, which by Eqn. (18) has leading term p/r. Fourth, for all three models, both the bias and variance tend to increase when the matrix is less rectangular, e.g., as p increases 10 to 100 for $n = 1000$. All in all, LEV is comparable to or outperforms UNIF, especially when the leverage scores are nonuniform.

4.3 Improvements from Shrinkage Leveraging and Unweighted Leveraging

Here, we will describe how our proposed SLEV and LEVUNW procedures can both lead to improvements over LEV and UNIF. Recall that LEV can lead to large MSE by inflating very small leverage scores. The SLEV procedure deals with this by considering a convex combination of the uniform distribution and the leverage score distribution, thereby providing a lower bound on the leverage scores; and the LEVUNW procedure deals with this by not rescaling the subproblem to be solved.

Consider Figures 4, 5, and 6, which present the variance and bias for synthetic data matrices (for GA, T_3, and T_1 data) of size $n \times p$, where $n = 1000$ and $p = 10$, 50, and 100, respectively. In each case, LEV, SLEV for three different values of the convex combination parameter α, and LEVUNW were considered. Several observations are worth making. First

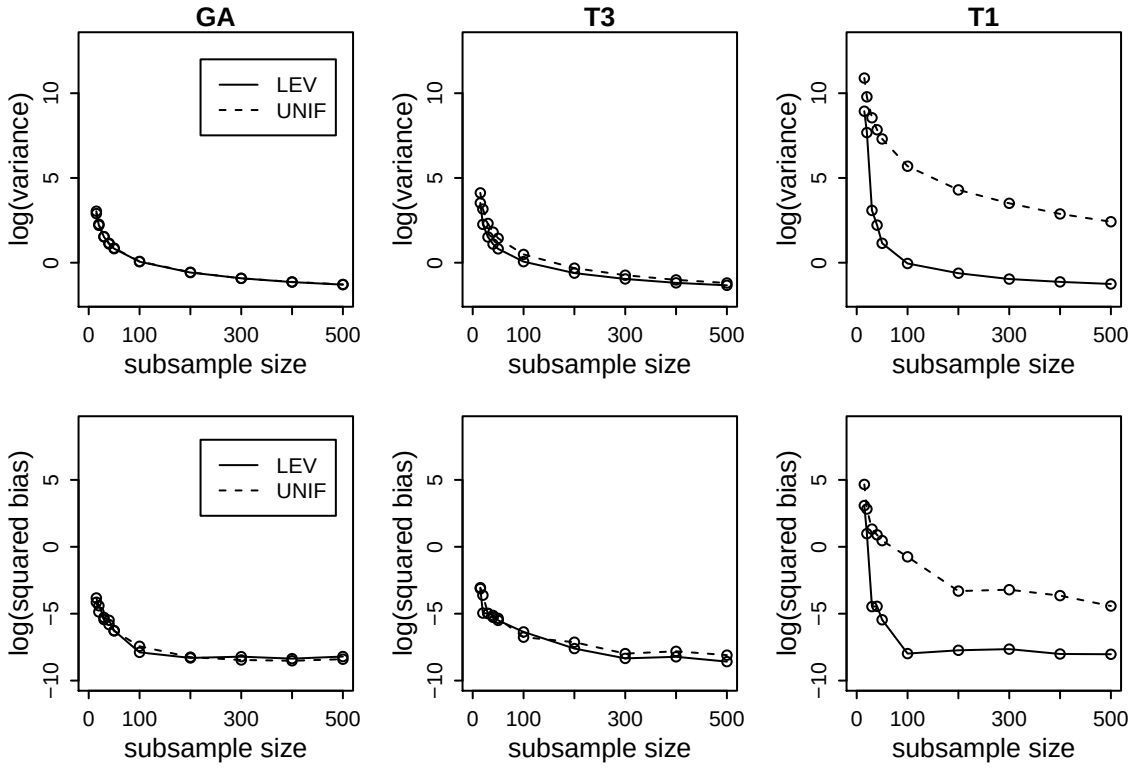


Figure 1: (Leveraging Versus Uniform Sampling subsection.) Comparison of variances and squared biases of the LEV and UNIF estimators in three data sets (GA, T_3, and T_1) for $n = 1000$ and $p = 10$. Left panels are GA data; Middle panels are T_3 data; Right panels are T_1 data. Upper panels are Logarithm of Variances; Lower panels are Logarithm of Squared bias. Black lines are LEV; Dash lines are UNIF.

of all, for GA data (left panel in these figures), all the results tend to be quite similar; but for T_3 data (middle panel) and even more so for T_1 data (right panel), differences appear. Second, SLEV with $\alpha \simeq 0.1$, i.e., when SLEV consists mostly of the uniform distribution, is notably worse in a manner similarly as with UNIF. Moreover, there is a gradual decrease in both bias and variance for our proposed SLEV as α is increased; and when $\alpha \simeq 0.9$ SLEV is slightly better than LEV. Finally, our proposed LEVUNW often has the smallest variance over a wide range of subsample sizes for both T_3 and T_1, although the effect is not major. All in all, these observations are consistent with our main theoretical results.

Next consider Figure 7. This figure examines the optimal convex combination choice for α in SLEV, with α being the x-axis in all the plots. Different column panels in Figure 7 correspond to different subsample sizes r. Recall that there are two conflicting goals for SLEV: adding $(1 - \alpha)/n$ to the small leverage scores will avoid substantially inflating the variance of the resulting estimate by samples with extremely small leverage scores; and doing so will lead to larger sample size r in order to obtain bounds of the form Eqns. (8) and (9). Figure 7 plots the variance and bias for T_1 data for a range of parameter values and for a

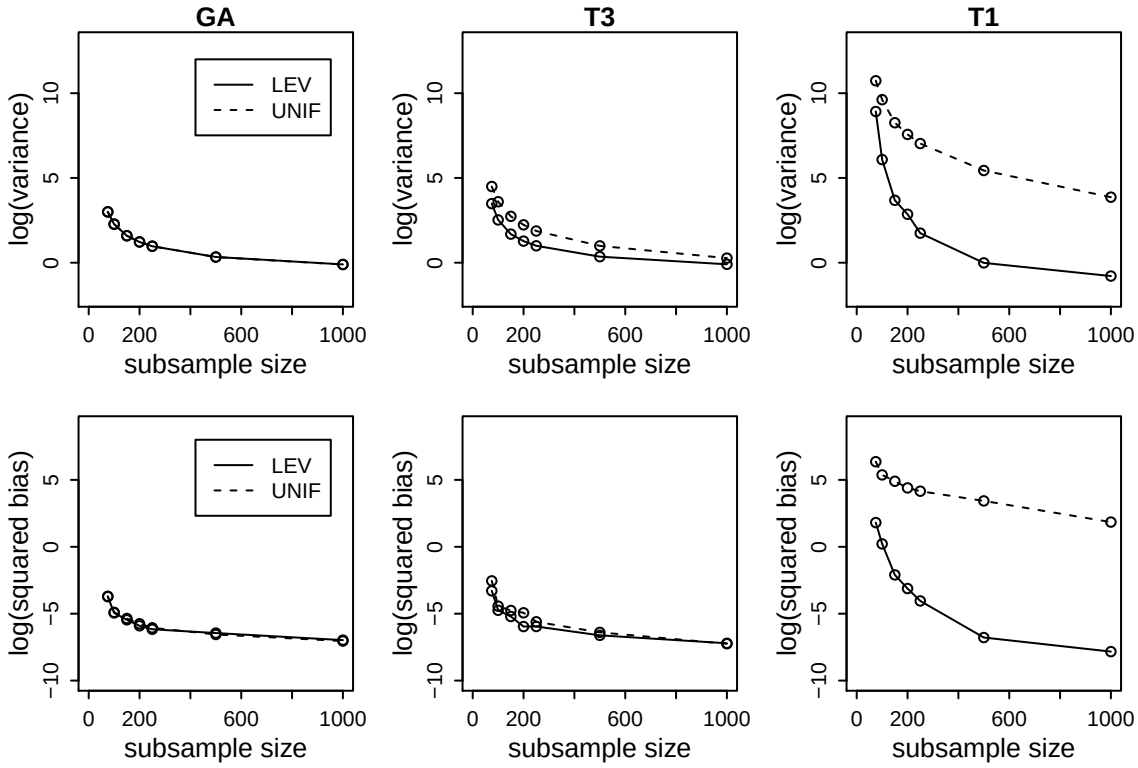


Figure 2: (Leveraging Versus Uniform Sampling subsection.) Same as Figure 1, except that $n = 1000$ and $p = 50$.

range of subsample sizes. In general, one sees that using SLEV to increase the probability of choosing small leverage components with α around $0.8 - 0.9$ (and relatedly shrinking the effect of large leverage components) has a beneficial effect on bias as well as variance. This is particularly true in two cases: first, when the matrix is very rectangular, e.g., when the $p = 10$, which is consistent with the leverage score statistics from Table 4.1; and second, when the subsample size r is larger, as the results for $r = 3p$ are much choppier (and for $r = 2p$, they are still choppier). As a rule of thumb, these plots suggest that choosing $\alpha = 0.9$, and thus using $\pi_i = \alpha\pi_i^{Lev} + (1 - \alpha)/n$ as the importance sampling probabilities, strikes a balance between needing more samples and avoiding variance inflation.

Inspecting in Figure 7 the grey lines, dots, and dashes, which correspond to LEVUNW for the various values of p, one can see that LEVUNW consistently has smaller variances than SLEV for all values of α. We should emphasize, though, that these are *unconditional* biases and variances. Since LEVUNW is approximately unbiased relative to the full sample *weighted* LS estimate $\hat{\boldsymbol{\beta}}_{wls}$, however, there is a large bias away from the full sample *unweighted* LS estimate $\hat{\boldsymbol{\beta}}_{ols}$. This suggests that LEVUNW may be used when the primary goal is to infer the true $\boldsymbol{\beta}_0$; but rather when the primary goal is to approximate the full sample unweighted LS estimate, or when *conditional* biases and variances are of interest,

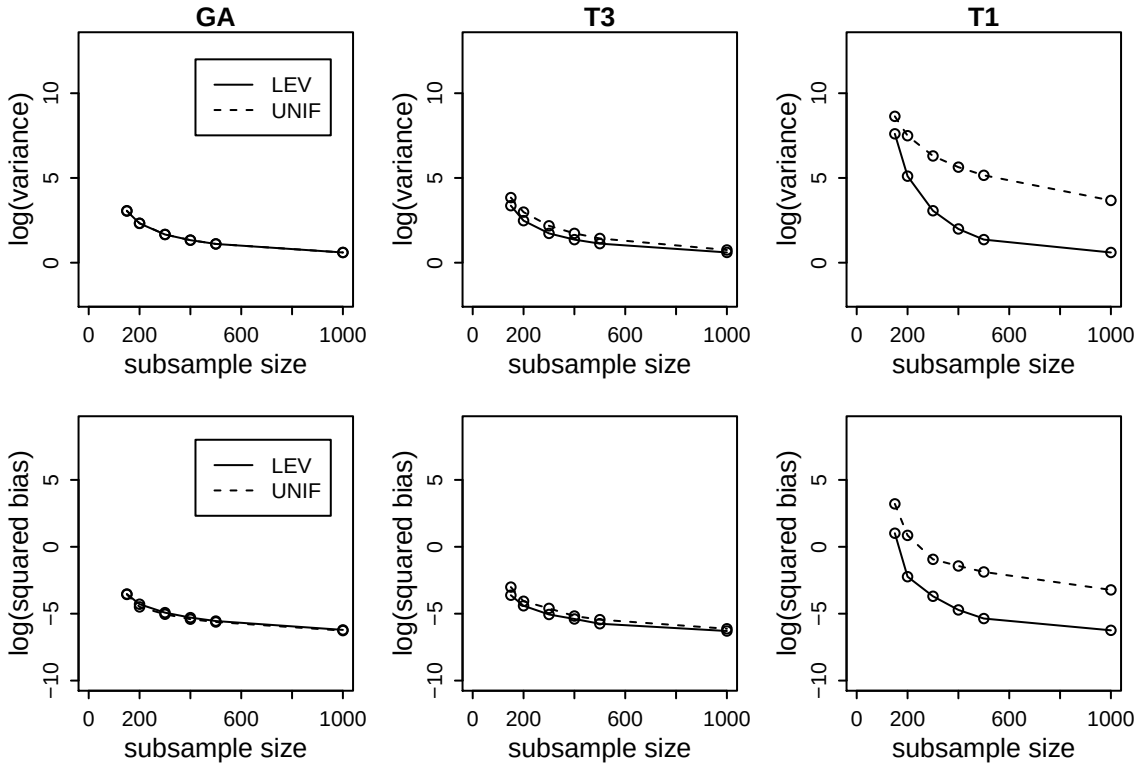


Figure 3: (Leveraging Versus Uniform Sampling subsection.) Same as Figure 1, except that $n = 1000$ and $p = 100$.

then SLEV may be more appropriate. We will discuss this in greater detail in Section 4.4 next.

4.4 Conditional Bias and Variance

Here, we will describe the properties of the *conditional* bias and variance under various subsampling estimators. These will provide a more direct comparison between Eqns. (14) and (15) from Lemma 2 and the corresponding results from Lemma 6. These will also provide a more direct comparison with previous work that has adopted an algorithmic perspective on algorithmic leveraging (Drineas et al., 2006; Mahoney, 2011; Drineas et al., 2012).

Consider Figure 8, which presents our main empirical results for conditional biases and variances. As before, matrices were generated from GA, T_3 and T_1; and we calculated the empirical bias and variance of UNIF, LEV, SLEV with $\alpha = 0.9$, and LEVUNW—in all cases, conditional on the empirical data $\boldsymbol{y}$. Several observations are worth making. First, for GA the variances are all very similar; and the biases are also similar, with the exception of LEVUNW. This is expected, since by the conditional expectation bounds from Lemma 6, LEVUNW is approximately unbiased, relative to the full sample *weighted* LS estimate $\hat{\boldsymbol{\beta}}_{wls}$—and thus there should be a large bias away from the full sample unweighted

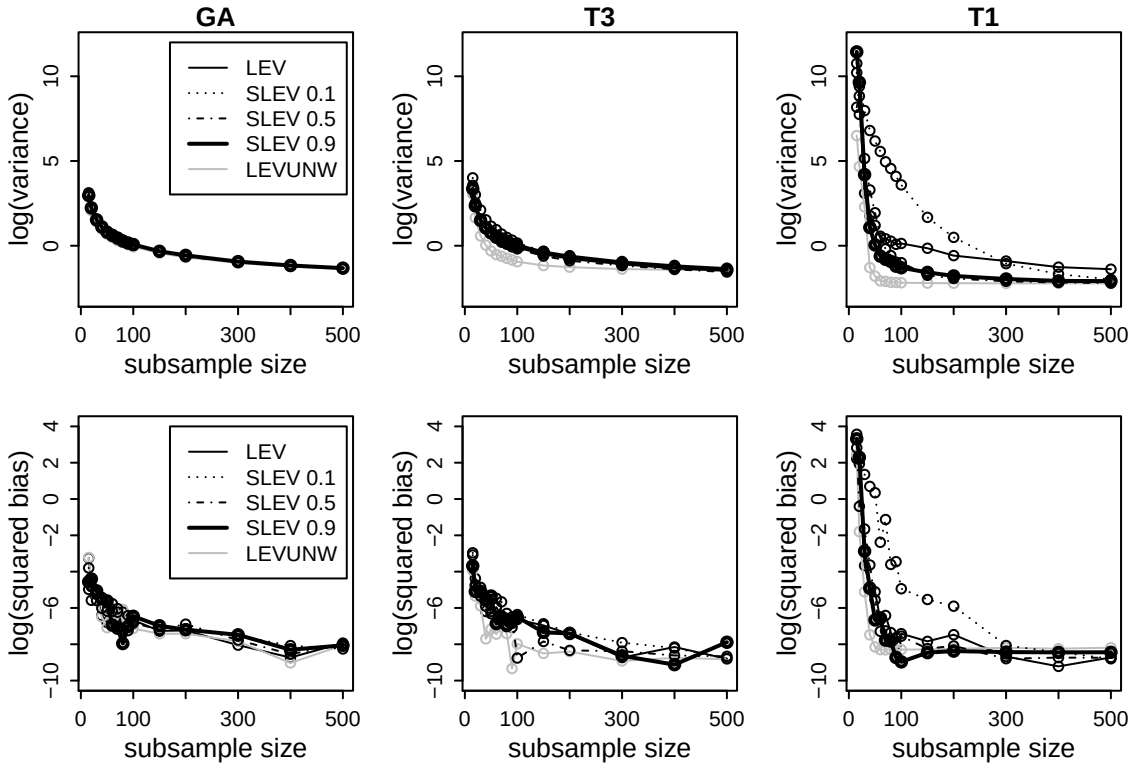


Figure 4: (Improvements from SLEV and LEVUNW subsection.) Comparison of variances and squared biases of the LEV, SLEV, and LEVUNW estimators in three data sets (GA, T_3, and T_1) for $n = 1000$ and $p = 10$. Left panels are GA data; Middle panels are T_3 data; Right panels are T_1 data. Grey lines are LEVUNW; black lines are LEV; dotted lines are SLEV with $\alpha = 0.1$; dot-dashed lines are SLEV with $\alpha = 0.5$; thick black lines are SLEV with $\alpha = 0.9$.

LS estimate. Second, for T_3 and even more prominently for T_1, the variance of LEVUNW is less than that for the other estimators. Third, when the leverage scores are very nonuniform, as with T_1, the relative merits of UNIF versus LEVUNW depend on the subsample size r. In particular, the bias of LEVUNW is larger than that of UNIF even for very aggressive downsampling; but it is substantially less than UNIF for moderate to large sample sizes.

Based on these and our other results, our default recommendation is to use SLEV (with either exact or approximate leverage scores) with $\alpha \approx 0.9$: it is no more than slightly worse than LEVUNW when considering unconditional biases and variances, and it can be much better than LEVUNW when considering conditional biases and variances.

5. Additional Empirical Evaluation

In this section, we provide additional empirical results (of a more specialized nature than those presented in Section 4). Here is a brief outline of the main results of this section.

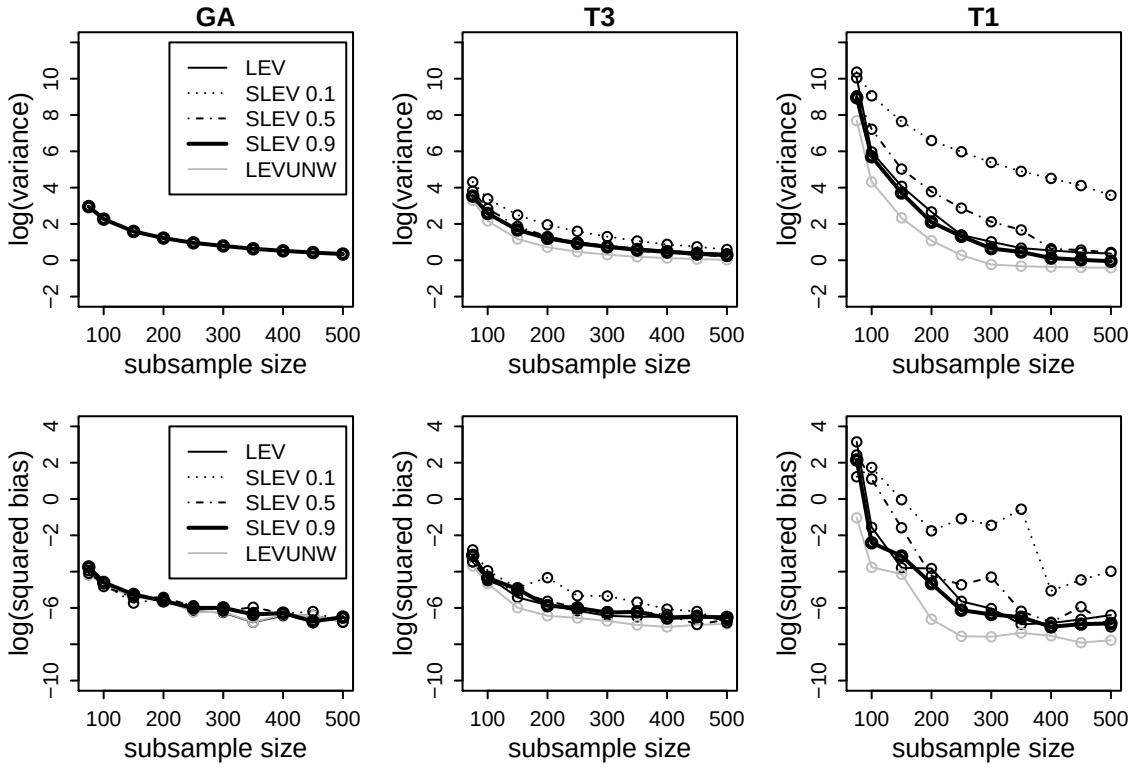


Figure 5: (Improvements from SLEV and LEVUNW subsection.) Same as Figure 4, except that $n = 1000$ and $p = 50$.

- In Section 5.1, we will consider the synthetic data, and we will describe what happens when the subsampled problem looses rank. This can happen if one is *extremely* aggressive in downsampling with SLEV; but it is much more common with UNIF, even if one samples many constraints. In both cases, the behavior of bias and variance is very different than when rank is preserved.

- Then, in Section 5.2, we will summarize our results on synthetic data when the leverage scores are computed approximately with the fast approximation algorithm of Drineas et al. (2012). Among other things, we will describe the running time of this algorithm, illustrating that it can solve larger problems compared to traditional deterministic methods; and we will evaluate the unconditional bias and variance of SLEV when this algorithm is used to approximate the leverage scores.

- Finally, in Section 5.3, we will consider real data, and we will present our results for the conditional bias and variance for two data sets that are drawn from our previous work in two genetics applications. One of these has very uniform leverage scores, and the other has moderately nonuniform leverage scores; and our results from the synthetic data hold also in these realistic applications.

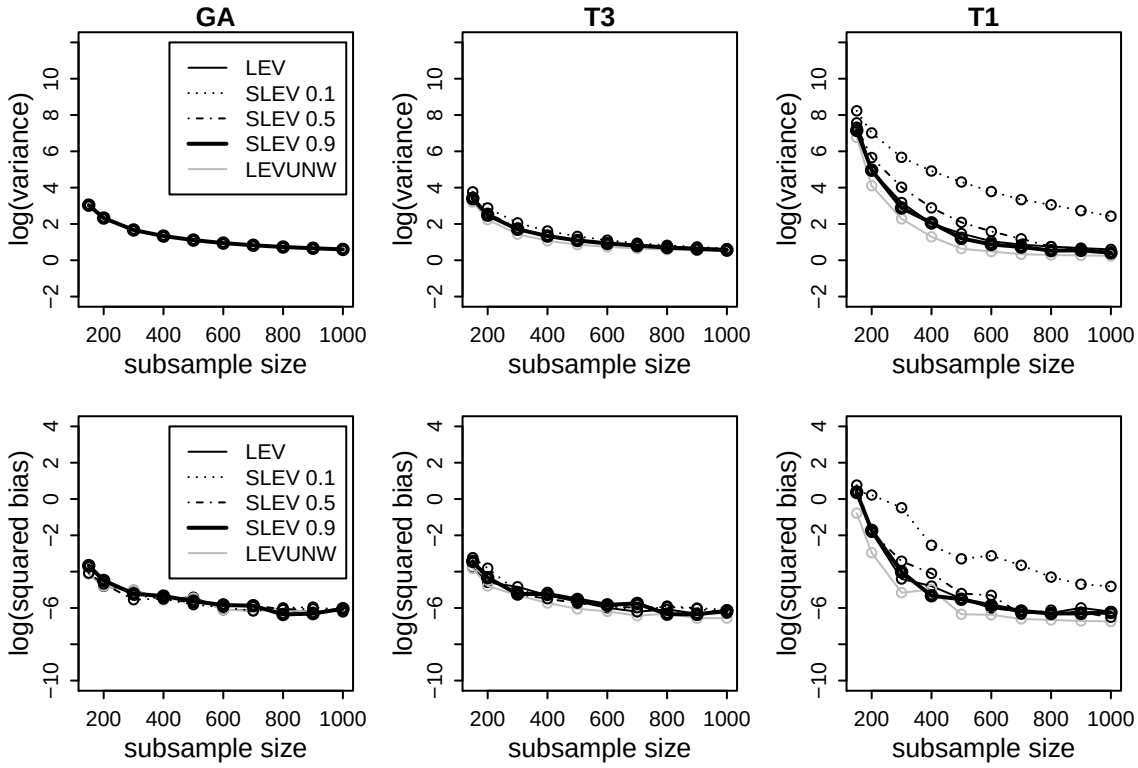


Figure 6: (Improvements from SLEV and LEVUNW subsection.) Same as Figure 4, except that $n = 1000$ and $p = 100$.

5.1 Leveraging and Uniform Estimates for Singular Subproblems

Here, we will describe the properties of LEV versus UNIF for situations in which rank is lost in the construction of the subproblem. That is, in some cases, the subsampled matrix, X^*, may have column rank that is smaller than the rank of the original matrix X, and this leads to a singular $X^{*T}X^* = X^TWX$. Of course, the LS solution of the subproblem can still be solved, but there will be a "bias" due to the dimensions that are not represented in the subsample. (We use the Moore-Penrose generalized inverse to compute the estimators when rank is lost in the construction of the subproblem.) Before describing these results, recall that algorithmic leveraging (in particular, LEV, but it holds for SLEV as well) guarantees that this will *not* happen in the following sense: if roughly $O(p \log p)$ rows of X are sampled using an importance sampling distribution that approximates the leverage scores in the sense of Eqn. (7), then with very high probability the matrix X^* does not lose rank (Drineas et al., 2006; Mahoney, 2011; Drineas et al., 2012). Indeed, this observation is crucial from the algorithmic perspective, i.e., in order to obtain relative-error bounds of the form of Eqns. (8) and (9), and thus it was central to the development of algorithmic leveraging. On the other hand, if one downsamples more aggressively, e.g., if one samples only, say, $p + 100$ or $p + 10$ rows, or if one uses uniform sampling when the leverage scores are very

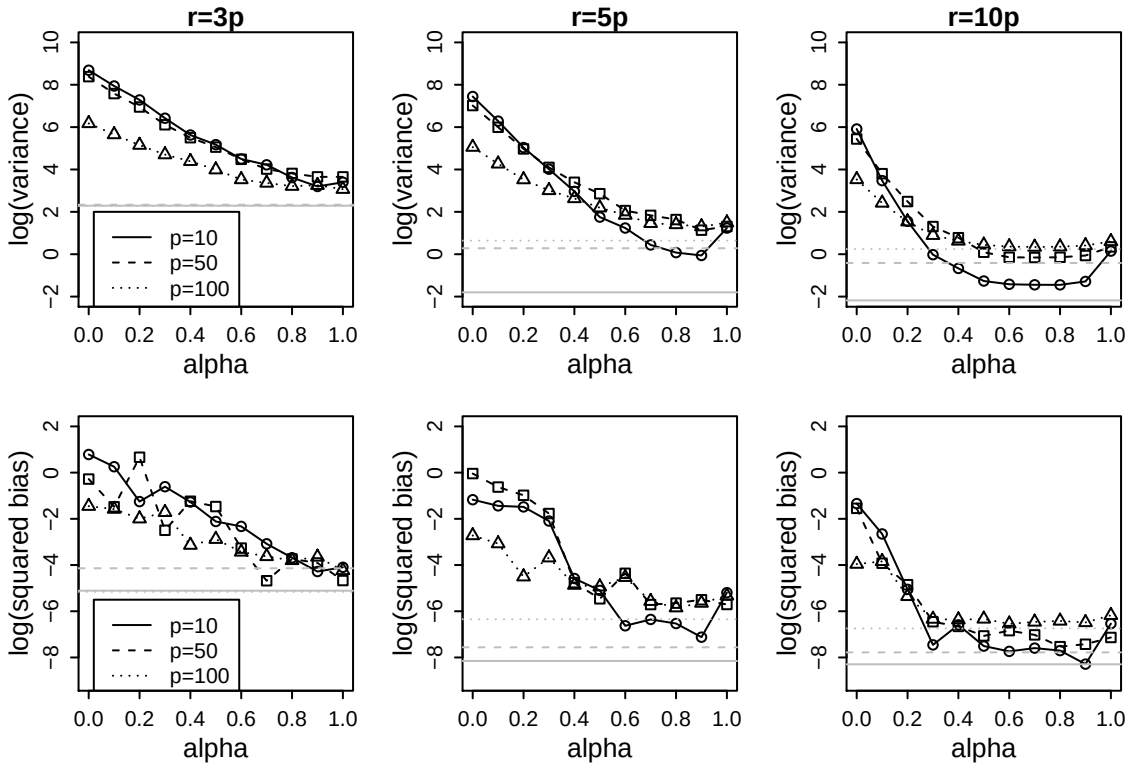


Figure 7: (Improvements from SLEV and LEVUNW subsection.) Varying α in SLEV. Comparison of variances and squared biases of the SLEV estimator in data generated from T_1 with $n = 1000$ and variable p. Left panels are subsample size $r = 3p$; Middle panels are $r = 5p$; Right panels are $r = 10p$. Circles connected by black lines are $p = 10$; squares connected by dash lines are $p = 50$; triangles connected by dotted lines are $p = 100$. Grey corresponds to the LEVUNW estimator.

nonuniform, then it is possible to lose rank. Here, we examine the statistical consequences of this.

We have observed this phenomenon with the synthetic data for both UNIF as well as for leverage-based sampling procedures; but the properties are somewhat different depending on the sampling procedure. To illustrate both of these with a single synthetic example, we first generated a 1000×10 matrix from multivariate t-distribution with 3 (or 2 or 1, denoted T_3, T_2, and T_1, respectively) degrees of freedom and covariance matrix $\Sigma_{ij} = 2 \times 0.5^{|i-j|}$; we then calculated the leverage scores of all rows; and finally we formed the matrix X was by keeping the 50 rows with highest leverage scores and replicating 950 times the row with the smallest leverage score. (This is a somewhat more realistic version of the toy **Worst-case Matrix** that is described in Appendix A) We then applied LEV and UNIF to the data sets with different subsample sizes, as we did for the results summarized in Section 4.2. Our results are summarized in Figure 9 and 10.

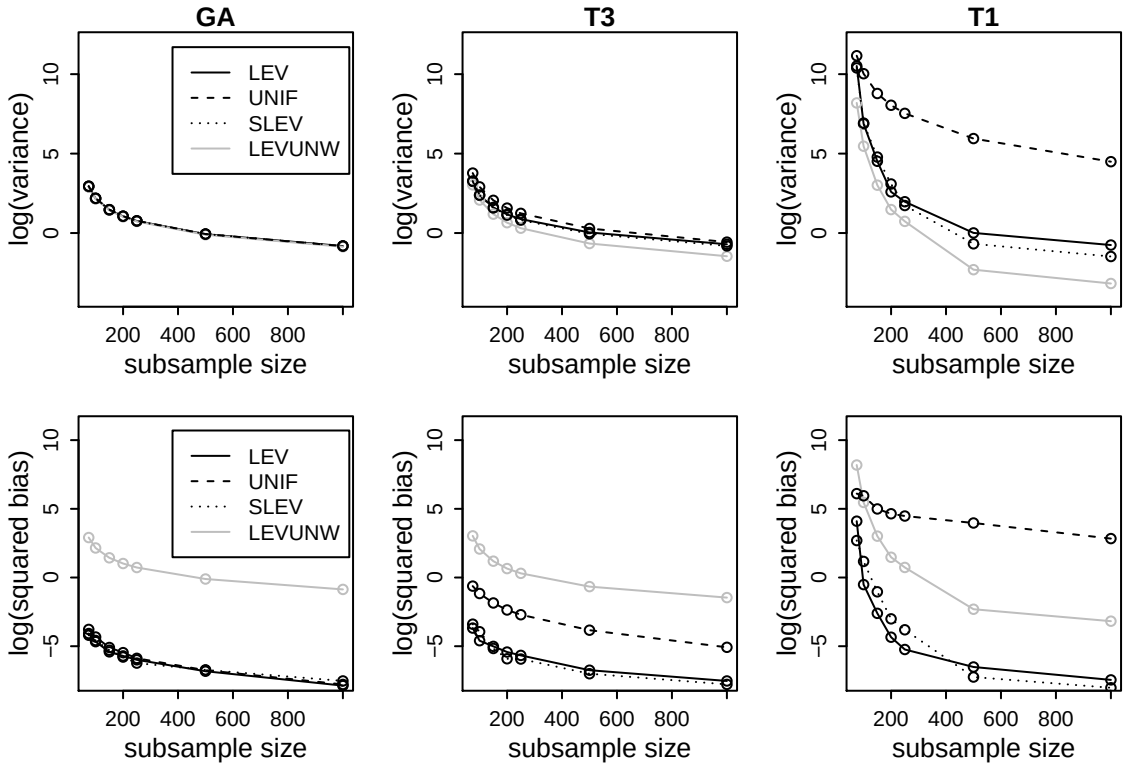


Figure 8: (Conditional Bias and Variance subsection.) Comparison of *conditional* variances and squared biases of the LEV and UNIF estimators in three data sets (GA, T_3, and T_1) for $n = 1000$ and $p = 50$. Left panels are GA data; Middle panels are T_3 data; Right panels are T_1 data. Upper panels are Variances; Lower panels are Squared Bias. Black lines for LEV estimate; dash lines for UNIF estimate; grey lines for LEVUNW estimate; dotted lines for SLEV estimate with $\alpha = 0.9$.

The top row of Figure 9 plots the fraction of singular $X^T W X$, out of 500 trials, for both LEV and UNIF; from left to right, results for T_3, T_2, and T_1 are shown. Several points are worth emphasizing. First, both LEV and UNIF loose rank if the downsampling is sufficiently aggressive. Second, for LEV, as long as one chooses more than roughly 20 (or less for T_2 and T_1), i.e., the ratio r/p is at least roughly 2, then rank is *not* lost; but for uniform sampling, one must sample a *much* larger fraction of the data. In particular, when fewer than $r = 100$ samples are drawn, nearly all of the subproblems constructed with the UNIF procedure are singular, and it is not until more than $r = 300$ that nearly all of the subproblems are not singular. Although these particular numbers depend on the particular data, one needs to draw many more samples with UNIF than with LEV in order to preserve rank and this is a very general phenomenon. The middle row of Figure 9 shows the boxplots of rank for the subproblem for LEV for those 500 tries; and the bottom row shows the boxplots of the rank of the subproblem for UNIF for those 500 tries. Note the unusual scale on the X-axis designed to highlight the lost rank data for both LEV as well

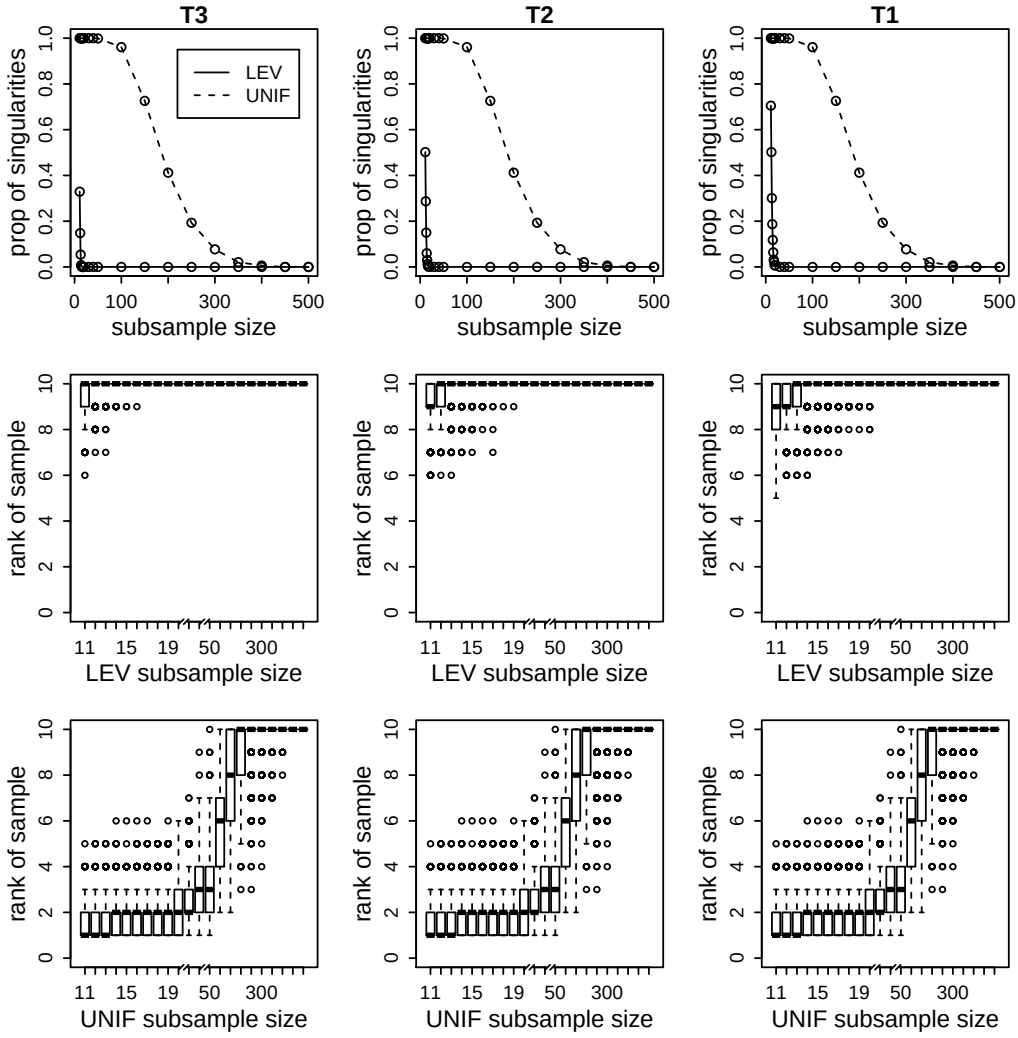


Figure 9: Comparison of LEV and UNIF when rank is lost in the sampling process ($n = 1000$ and $p = 10$ here). Left panels are T_3; Middle panels are T_2; Right panels are T_1. Upper panels are proportion of singular $X^T W X$, out of 500 trials, for both LEV (solid lines) and UNIF (dashed lines); Middle panels are boxplots of ranks of 500 LEV subsamples; Lower panels are boxplots of ranks of 500 UNIF subsamples. Note the nonstandard scaling of the X axis.

as UNIF. These boxplots illustrate the sigmoidal distribution of ranks obtained by UNIF as a function of the number of samples and the less severe beginning of the sigmoid for LEV; and they also show that when subproblems are singular, then often many dimensions fail to be captured. All in all, LEV outperforms UNIF, especially when the leverage scores are nonuniform.

Figure 10 illustrates the variance and bias of the corresponding estimators. In particular, the upper panels plot the logarithm of variances; the middle panels plot the same quantities, except that it is zoomed-in on the X-axis; and the lower panels plot the logarithm of

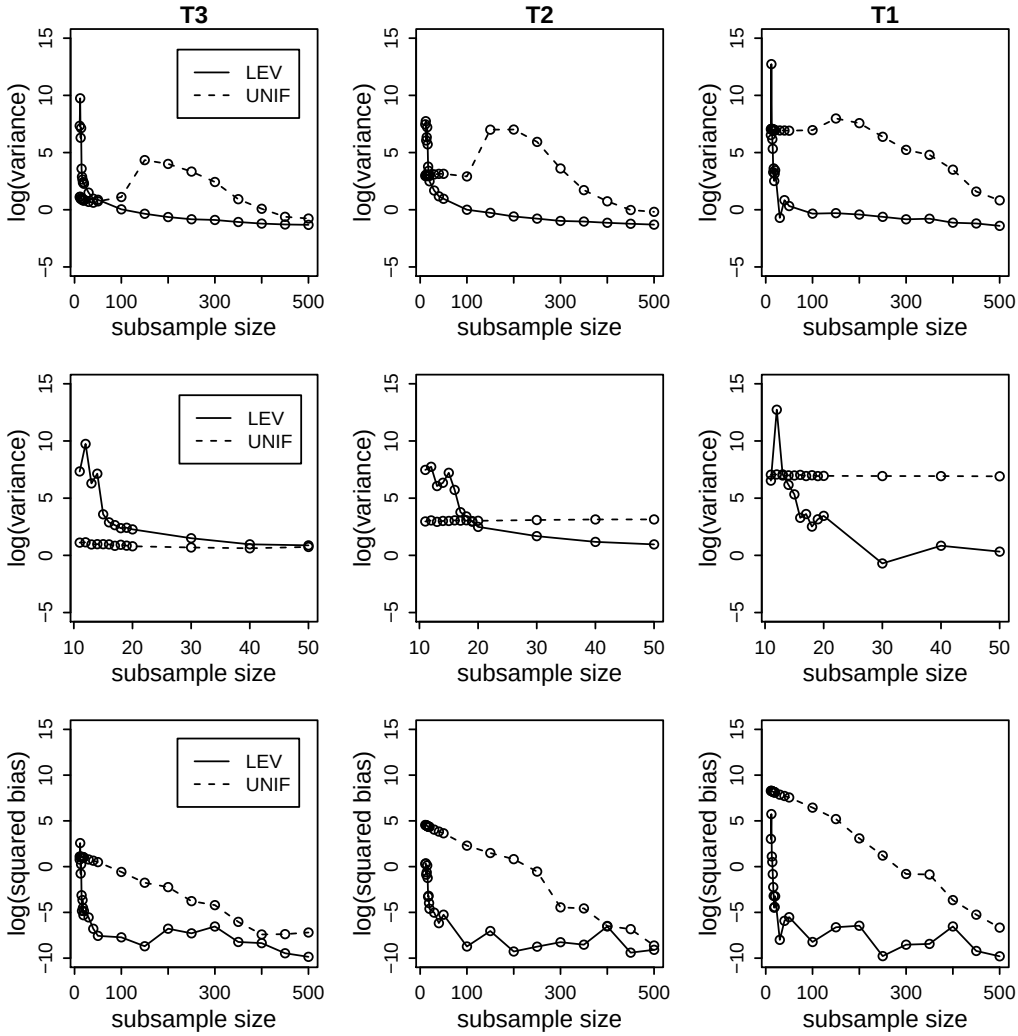


Figure 10: Comparison of LEV and UNIF when rank is lost in the sampling process ($n = 1000$ and $p = 10$ here). Left panel are T_3; Middle panels are T_2; Right panels are T_1. Upper panels are logarithm of variances of the estimates; Middle panels are logarithm of variances, zoomed-in on the X-axis; Lower panels are logarithm of squared bias of the estimates. Black line for LEV; Dash line for UNIF.

squared bias. As before, the left/middle/right panels present results for the $T_3/T_2/T_1$ data, respectively. The behavior here is very different that that shown in Figures 1, 2, and 3; and several observations are worth making. First, for all three models and for both LEV and UNIF, when the downsampling is very aggressive, e.g, $r = p+5$ or $r = p+10$, then the bias is comparable to the variance. That is, since the sampling process has lost dimensions, the linear approximation implicit in our Taylor expansion is violated. Second, both bias and variance are worse for T_1 than for T_2 than for T_3, which is consistent with Table 4.1, but the effect is minor; and the bias and variance are generally much worse for UNIF than for LEV. Third, as r increases, the variance for UNIF increases, hits a maximum and then decreases;

and at the same time the bias for UNIF gradually decreases. Upon examining the original data, the reason that there is very little variance initially is that most of the subsamples have rank 1 or 2; then the variance increases as the dimensionality of the subsamples increases; and then the variance decreases due to the $1/r$ scaling, as we saw in the plots in Section 4.2. Fourth, as r increases, both the variance and bias of LEV decrease, as we saw in Section 4.2; but in the aggressive downsampling regime, i.e., when r is very small, the variance of LEV is particularly "choppy," and is actually worse than that of UNIF, perhaps also due to rank deficiency issues.

5.2 Approximate Leveraging via the Fast Leveraging Algorithm

Here, we employ the fast randomized algorithm from Drineas et al. (2012) to compute approximations to the leverage scores of X, to be used in place of the exact leverage scores in LEV, SLEV, and LEVUNW. To start, we provide a brief description of the algorithm of Drineas et al. (2012), which takes as input an arbitrary $n \times p$ matrix X.

- Generate an $r_1 \times n$ random matrix Π_1 and a $p \times r_2$ random matrix Π_2.
- Let R be the R matrix from a QR decomposition of $\Pi_1 X$.
- Compute and return the leverage scores of the matrix $XR^{-1}\Pi_2$.

For appropriate choices of r_1 and r_2, if one chooses Π_1 to be a Hadamard-based random projection matrix, then this algorithm runs in $o(np^2)$ time, and it returns $1 \pm \epsilon$ approximations to all the leverage scores of X (Drineas et al., 2012). In addition, with a high-quality implementation of the Hadamard-based random projection, this algorithm runs faster than traditional deterministic algorithms based on Lapack for matrices as small as several thousand by several hundred (Avron et al., 2010; Gittens and Mahoney, 2013).

We have implemented in the software environment R two variants of this fast algorithm of Drineas et al. (2012), and we have compared it with QR-based deterministic algorithms also supported in R for computing the leverage scores exactly. In particular, the following results were obtained on a PC with Intel Core i7 Processor and 6 Gbytes RAM running Windows 7, on which we used the software package R, version 2.15.2. In the following, we refer to the above algorithm as BFast (the Binary Fast algorithm) when (up to normalization) each element of Π_1 and Π_2 is generated i.i.d. from $\{-1, 1\}$ with equal sampling probabilities; and we refer to the above algorithm as GFast (the Gaussian Fast algorithm) when each element of Π_1 is generated i.i.d. from a Gaussian distribution with mean zero and variance $1/n$ and each element of Π_2 is generated i.i.d. from a Gaussian distribution with mean zero and variance $1/p$. In particular, note that here we do not consider Hadamard-based projections for Π_1 or more sophisticated parallel and distributed implementations of these algorithms (Avron et al., 2010; Meng et al., 2014; Gittens and Mahoney, 2013; Yang et al., 2013).

To illustrate the behavior of this algorithm as a function of its parameters, we considered synthetic data where the $20,000 \times 1,000$ design matrix X is generated from T_1 distribution. All the other parameters are set to be the same as before, except $\Sigma_{ij} = 0.1$, for $i \neq j$, and $\Sigma_{ii} = 2$. We then applied BFast and GFast with varying r_1 and r_2 to the data. In particular, we set $r_1 = p, 1.5p, 2p, 3p, 5p$, where $p = 1,000$, and we set $r_2 = \kappa \log(n)$, for $\kappa =$

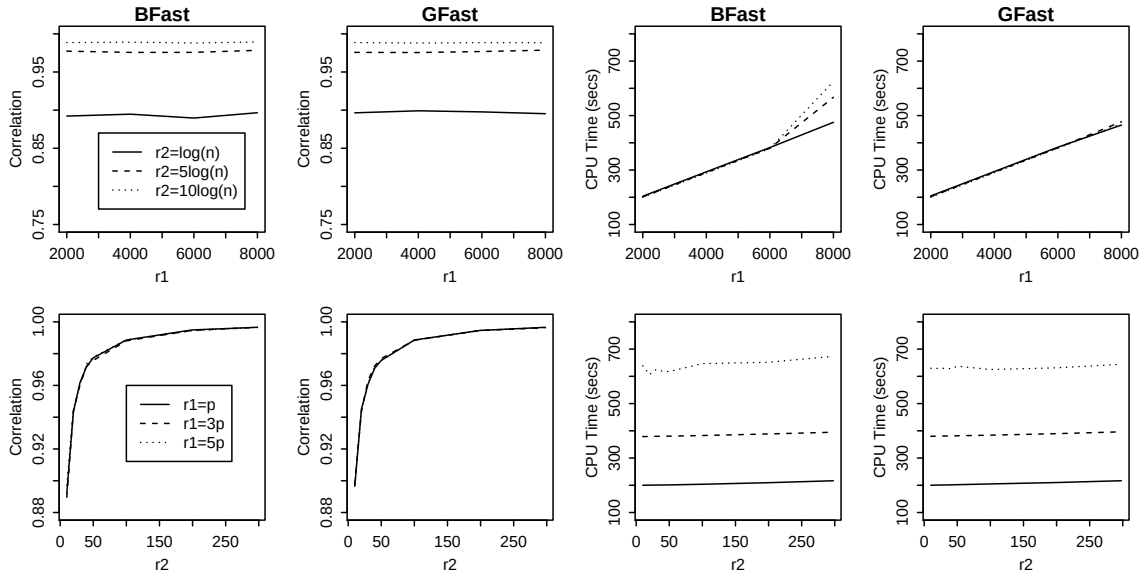


Figure 11: (Fast Leveraging Algorithm subsection.) Effect of approximating leverage scores using BFast and GFast for varying parameters. Upper panels: Varying parameter r_1 for fixed r_2, where $r_2 = \log(n)$ (black lines), $r_2 = 5\log(n)$ (dashed lines), and $r_2 = 10\log(n)$ (dotted lines). Lower panels: Varying parameter r_2 for fixed r_1, where $r_1 = p$ (black lines), $r_1 = 3p$ (dashed lines), and $r_1 = 5p$ (dotted lines). Left two panels: Correlation between exact leverage scores and leverage scores approximated using BFast and GFast, for varying r_1 and r_2. Right two panels: CPU time for varying r_1 and r_2, using BFast and GFast.

$1, 2, 3, 4, 5, 10, 20$, where $n = 20,000$. See Figure 11, which presents both a summary of the correlation between the approximate and exact leverage scores as well as a summary of the running time for computing the approximate leverage scores, as r_1 and r_2 are varied for both BFast and GFast. We can see that the correlations between approximated and exact leverage scores are not very sensitive to r_1, whereas the running time increases roughly linearly for increasing r_1. In contrast, the correlations between approximated and exact leverage scores increases rapidly for increasing r_2, whereas the running time does not increase much when r_2 increases. These observations suggest that we may use a combination of small r_1 and large r_2 to achieve high-quality approximation and short running time.

Next, we examine the running time of the approximation algorithms for computing the leverage scores. Our results for running times are summarized in Figure 12. In that figure, we plot the running time as sample size n and predictor size p are varied for BFast and GFast. We can see that when the sample size is very small, the computation time of the fast algorithms is slightly worse than that of the exact algorithm. (This phenomenon occurs primarily due to the fact that the fast algorithm requires additional projection and matrix multiplication steps, which dominate the running time for very small matrices.) On the other hand, when the sample size is larger than ca. $20,000$, the computation time of the

fast approximation algorithms becomes slightly less expensive than that of exact algorithm. Much more significantly, when the sample size is larger than roughly $35,000$, the exact algorithm requires more memory than our standard R environment can provide, and thus it fails to run at all. In contrast, the fast algorithms can work with sample size up to roughly $60,000$.

That is, the use of this randomized algorithm to approximate the leverage scores permits us to work with data that are roughly 1.5 times larger in n or p, even when a simple vanilla implementation is provided in the R environment. If one is interested in much larger inputs, e.g., with $n = 10^6$ or more, then one should probably not work within R and instead use Hadamard-based random projections for Π_1 and/or the use of more sophisticated methods, such as those described in Avron et al. (2010); Meng et al. (2014); Gittens and Mahoney (2013); Yang et al. (2013); here we simply evaluate an implementation of these methods in R. The reason that BFast and GFast can run for much larger input is likely that the computational bottleneck for the exact algorithm is a QR decomposition, while the computational bottleneck for the fast randomized algorithms is the matrix-matrix multiplication step.

Finally, we evaluate the bias and variance of LEV, SLEV and LEVUNW estimates where the leverage scores are calculated using exact algorithm, BFast, and GFast. In Figure 13, we plot the variance and squared bias for T_3 data sets. (We have observed similar but slightly smoother results for the Gaussian data sets and similar but slightly choppier results for the T_1 data sets.) Observe that the variances of LEV estimates where the leverage scores are calculated using exact algorithm, BFast, and GFast are almost identical; and this observation is also true for SLEV and LEVUNW estimates. All in all, using the fast approximation algorithm of Drineas et al. (2012) to compute approximations to the leverage scores for use in LEV, SLEV, and LEVUNW leads to improved algorithmic performance, while achieving nearly identical statistical results as LEV, SLEV, and LEVUNW when the exact leverage scores are used.

5.3 Illustration of the Method on Real Data

Here, we provide an illustration of our methods on two real data sets drawn from two problems in genetics with which we have prior experience (Dalpiaz et al., 2013; Mahoney and Drineas, 2009). The first data set has relatively uniform leverage scores, while the second data set has somewhat more nonuniform leverage scores. These two examples simply illustrate that observations we made on the synthetic data also hold for more realistic data that we have studied previously. For more information on the application of these ideas in genetics, see previous work on PCA-correlated SNPs (Paschou et al., 2007, 2010).

5.3.1 Linear Model for Bias Correction in RNA-Seq Data

In order to illustrate how our methods perform on a real data set with nearly uniform leverage scores, we consider an RNA-Seq data set containing $n = 51,751$ read counts from embryonic mouse stem cells (Cloonan et al., 2008). Recall that RNA-Seq is becoming the major tool for transcriptome analysis; it produces digital signals by obtaining tens of millions of short reads; and after being mapped to the genome, RNA-Seq data can be summarized by a sequence of short-read counts. Recent work found that short-read counts have significant

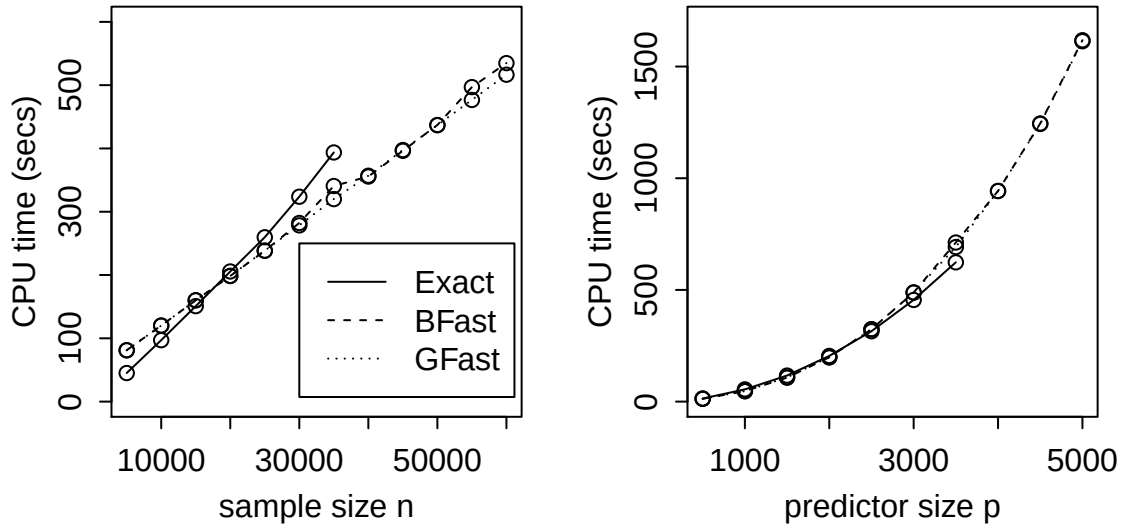


Figure 12: (Fast Leveraging Algorithm subsection.) CPU time for calculating exact leverage scores and approximate leverage scores using the BFast and GFast versions of the fast algorithm of (Drineas et al., 2012). Left panel is CPU time for varying sample size n for fixed predictor size $p = 500$; Right panel is CPU time for varying predictor size p for fixed sample size $n = 2000$. Black lines connect the CPU time for calculating exact leverage scores; dash lines connect the CPU time for using GFast to approximate the leverage scores; dotted lines connect the CPU time for using BFast to approximate the leverage scores.

sequence bias (Li et al., 2010). Here, we consider a simplified linear model of Dalpiaz et al. (2013) for correcting sequence bias in RNA-Seq. Let n_{ij} denote the counts of reads that are mapped to the genome starting at the jth nucleotide of the ith gene, where $i = 1, 2, \ldots, 100$ and $j = 1, \ldots, L_i$. We assume that the log transformed count of reads, $y_{ij} = \log(n_{ij} + 0.5)$, depends on 40 nucleotides in the neighborhood, denoted as $b_{ij,-20}, b_{ij,-19}, \ldots, b_{ij,18}, b_{ij,19}$ through the following linear model: $y_{ij} = \alpha + \sum_{k=-20}^{19} \sum_{h \in \mathcal{H}} \beta_{kh} I(b_{ij,k} = h) + \epsilon_{ij}$, where $\mathcal{H} = \{A, C, G\}$, where T is used as the baseline level, α is the grand mean, $I(b_{ij,k} = h)$ equals to 1 if the kth nucleotide of the surrounding sequence is h, and 0 otherwise, β_{kh} is the coefficient of the effect of nucleotide h occurring in the kth position, and $\epsilon_{ij} \sim N(0, \sigma^2)$. This linear model uses $p = 121$ parameters to model the sequence bias of read counts. For $n = 51,751$, model-fitting via LS is time-consuming.

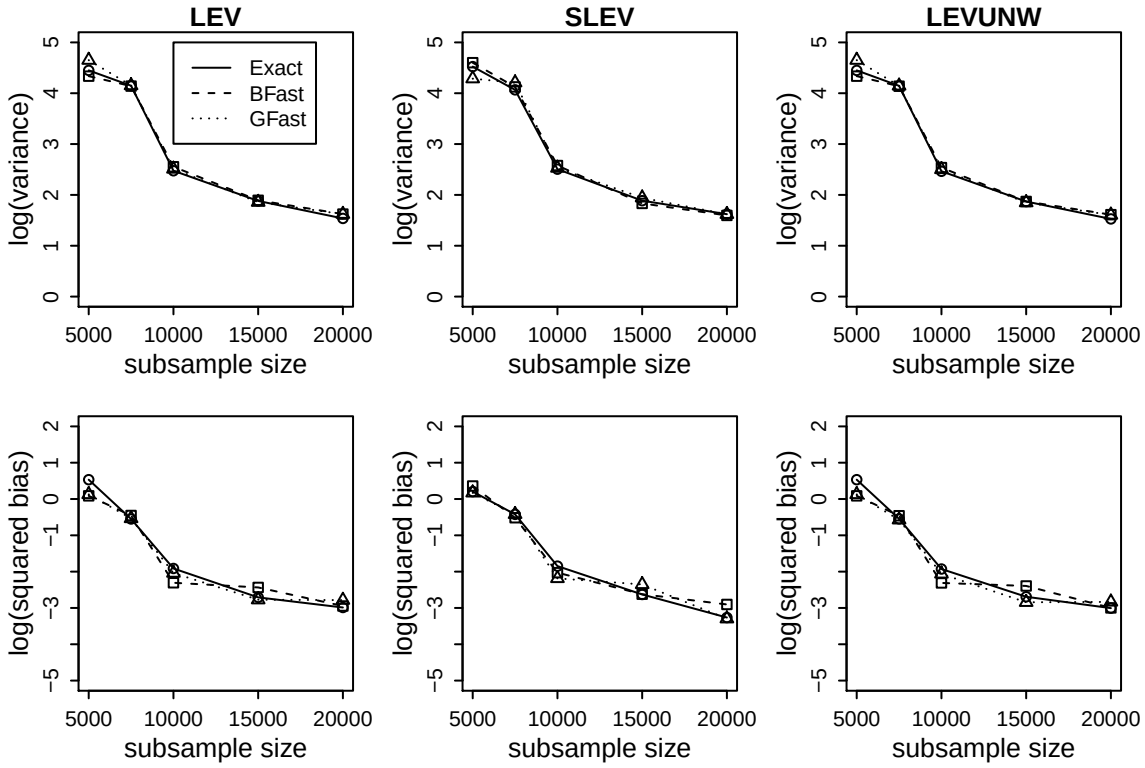


Figure 13: (Fast Leveraging Algorithm subsection.) Comparison of variances and squared biases of the LEV, SLEV, and LEVUNW estimators in T_3 data sets for $n = 20000$ and $p = 5000$ using BFast and GFast versions of the fast algorithm of (Drineas et al., 2012). Left panels are LEV estimates; Middle panels are SLEV estimates; Right panels are LEVUNW estimates. Black lines are exact algorithm; dash lines are BFast; dotted lines are GFast.

Coefficient estimates were obtained using three subsampling algorithms for seven different subsample sizes: $2p, 3p, 4p, 5p, 10p, 20p, 50p$. We compare the estimates using the sample bias and variances; and, for each subsample size, we repeat our sampling 100 times to get 100 estimates. (At each subsample size, we take one hundred subsamples and calculate all the estimates; we then calculate the bias of the estimates with respect to the full sample least squares estimate and their variance.) See Figure 14 for a summary of our results. In the left panel of Figure 14, we plot the histogram of the leverage score sampling probabilities. Observe that the distribution is quite uniform, suggesting that leverage-based sampling methods will perform similarly to uniform sampling. To demonstrate this, the middle and right panels of Figure 14 present the (conditional) empirical variances and biases of each of the four estimates, for seven different subsample sizes. Observe that LEV, LEVUNW, SLEV, and UNIF all have comparable sample variances. When the subsample size is very small, all four methods have comparable sample bias; but when the subsample size is larger, then LEVUNW has a slightly larger bias than the other three estimates.

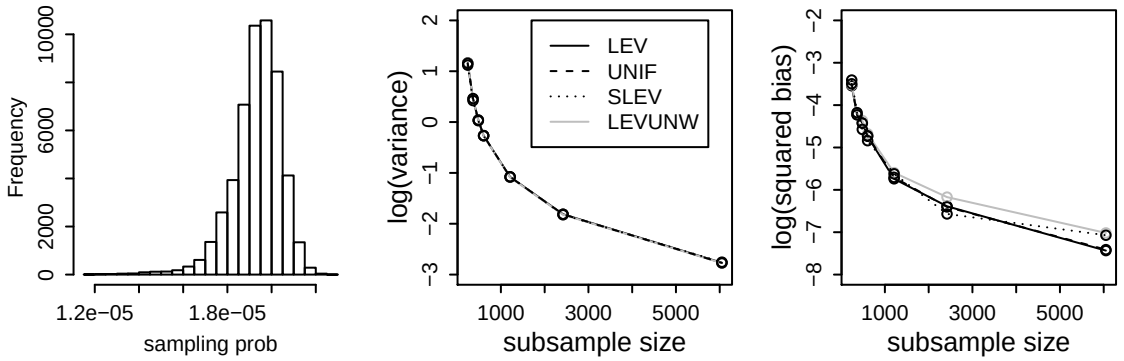


Figure 14: Empirical results for real data. Left panel is the histogram of the leverage score sampling probabilities for the RNA-Seq data (the largest leverage score is 2.25×10^{-5}, and the mean is 1.93×10^{-5}, i.e., the largest is only slightly larger than the mean); Middle panel is the empirical *conditional* variances of the LEV, UNIF, LEVUNW, and SLEV estimates; Right panel is the empirical *conditional* biases. Black lines for LEV; dash lines for UNIF; grey lines for LEVUNW; dotted lines for SLEV with $\alpha = 0.9$.

5.3.2 Linear Model for Predicting Gene Expressions of Cancer Patient

In order to illustrate how our methods perform on real data with moderately nonuniform leverage scores, we consider a microarray data set that was presented in Nielsen et al. (2002) (and also considered in Mahoney and Drineas 2009) for 46 cancer patients with respect to $n = 5,520$ genes. Here, we randomly select one patient's gene expression as the response $\boldsymbol{y}$ and use the remaining patients' gene expressions as the predictors (so $p = 45$); and we predict the selected patient's gene expression using other patients gene expressions through a linear model. We fit the linear model using subsampling algorithms with nine different subsample sizes. See Figure 15 for a summary of our results. In the left panel of Figure 15, we plot the histogram of the leverage score sampling probabilities. Observe that the distribution is highly skewed and quite a number of probabilities are significantly larger than the average probability. Thus, one might expect that leveraging estimates will have an advantage over the uniform sampling estimate. To demonstrate this, the middle and right panels of Figure 15 present the (conditional) empirical variances and biases of each of the four estimates, for nine different subsample sizes. Observe that SLEV and LEV have smaller sample variance than LEVUNW and that UNIF consistently has the largest variance. Interestingly, since LEVUNW is approximately unbiased to the weighted least squares estimate, here we observe that LEVUNW has by far the largest bias and that the bias does not decrease as the subsample size increases. In addition, when the subsample size is less than 2000, the biases of LEV, SLEV and UNIF are comparable; but when the subsample size is greater than 2000, LEV and SLEV have slightly smaller bias than UNIF.

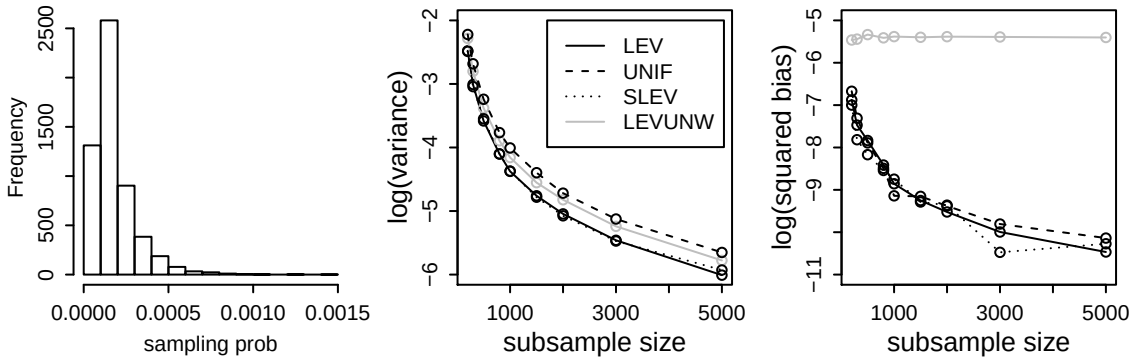


Figure 15: Empirical results for real data. Left panel is the histogram of the leverage score sampling probabilities for the microarray data (the largest leverage score is 0.00124, and the mean is 0.00018, i.e., the largest is 7 times the mean); Middle panel is the empirical *conditional* variances of the LEV, UNIF, LEVUNW, and SLEV estimates; Right panel is the empirical *conditional* biases. Black lines for LEV; dash lines for UNIF; grey lines for LEVUNW; dotted lines for SLEV with $\alpha = 0.9$.

6. Discussion and Conclusion

Algorithmic leveraging—a recently-popular framework for solving large least-squares regression and other related matrix problems via sampling based on the empirical statistical leverage scores of the data—has been shown to have many desirable *algorithmic* properties. In this paper, we have adopted a *statistical* perspective on algorithmic leveraging, and we have demonstrated how this leads to improved performance of this paradigm on real and synthetic data. In particular, from the algorithmic perspective of worst-case analysis, leverage-based sampling provides uniformly superior worst-case algorithmic results, when compared with uniform sampling. Our statistical analysis, however, reveals that, from the statistical perspective of bias and variance, neither leverage-based sampling nor uniform sampling dominates the other. Based on this, we have developed new statistically-inspired leveraging algorithms that achieve improved statistical performance, while maintaining the algorithmic benefits of the usual leverage-based method. Our empirical evaluation demonstrates that our theory is a good predictor of the practical performance of both existing as well as our newly-proposed leverage-based algorithms. In addition, our empirical evaluation demonstrates that, by using a recently-developed algorithm to approximate the leverage scores, we can compute improved approximate solutions for much larger least-squares problems than we can compute the exact solutions with traditional deterministic algorithms.

Finally, we should note that, while our results are straightforward and intuitive, obtaining them was not easy, in large part due to seemingly-minor differences between problem formulations in statistics, computer science, machine learning, and numerical linear algebra. Now that we have "bridged the gap" by providing a statistical perspective on a recently-

popular algorithmic framework, we expect that one can ask even more refined statistical questions of this and other related algorithmic frameworks for large-scale computation.

Acknowledgments

This research is supported by NSF grant CDS&E-MSS 1228288(1440038) to PM, 1228155 to MWM, 1228246 to BY. PM acknowledges partial research supports from NSF grants DMS-1055815 (DMS-1438957) and DMS-1222718 (DMS-1440037). MWM acknowledges partial research supports from the Army Research Office, the Defense Advanced Research Projects Agency, and the National Science Foundation. BY acknowledges partial research supports from NSF grants DMS-1107000 and DMS-1160319, ARO grant W911NF-11-1-0114, and the Center for Science of Information (CSoI), an NSF Science and Technology Center, under grant agreement CCF-0939370.

Appendix A. Asymptotic Analysis and Toy Data

In this appendix, we will relate our analytic methods to the notion of asymptotic relative efficiency, and we will consider several toy data sets that illustrate various aspects of algorithmic leveraging. Although the results of this appendix are not used elsewhere, and thus some readers may prefer skip this appendix, we include it in order to relate our approach to ideas that may be more familiar to certain readers.

A.1 Asymptotic Relative Efficiency Analysis

Here, we present an asymptotic analysis comparing UNIF with LEV, SLEV, and LEVUNW in terms of their relative efficiency. Recall that one natural way to compare two procedures is to compare the sample sizes at which the two procedures meet a given standard of performance. One such standard is efficiency, which addresses how "spread out" about $\boldsymbol{\beta}_0$ is the estimator. In this case, the smaller the variance, the more "efficient" is the estimator (Serfling, 2010). Since $\boldsymbol{\beta}_0$ is a p-dimensional vector, to determine the relative efficiency of two estimators, we consider the linear combination of $\boldsymbol{\beta}_0$, i.e., $c^T\boldsymbol{\beta}_0$, where c is the linear combination coefficient. In somewhat more detail, when $\hat{\boldsymbol{\beta}}$ and $\tilde{\boldsymbol{\beta}}$ are two one-dimensional estimates, their relative efficiency can be defined as

$$e(\hat{\boldsymbol{\beta}}, \tilde{\boldsymbol{\beta}}) = \frac{\mathrm{Var}(\tilde{\boldsymbol{\beta}})}{\mathrm{Var}(\hat{\boldsymbol{\beta}})},$$

and when $\hat{\boldsymbol{\beta}}$ and $\tilde{\boldsymbol{\beta}}$ are two p-dimensional estimates, we can take their linear combinations $c^T\hat{\boldsymbol{\beta}}$ and $c^T\tilde{\boldsymbol{\beta}}$, where c is the linear combination coefficient vector, and define their relative efficiency as

$$e(c^T\hat{\boldsymbol{\beta}}, c^T\tilde{\boldsymbol{\beta}}) = \frac{\mathrm{Var}(c^T\tilde{\boldsymbol{\beta}})}{\mathrm{Var}(c^T\hat{\boldsymbol{\beta}})}.$$

In order to discuss asymptotic relative efficiency, we start with the following seemingly-technical observation.

Definition 7 *A $k \times k$ matrix A is said to be $A = O(\alpha_n)$ if and only if every element of A satisfies $A_{ij} = O(\alpha_n)$ for $i, j = 1, \ldots, k$.*

Assumption 1 *$X^T X = \sum_{i=1}^n \boldsymbol{x}_i \boldsymbol{x}_i^T$ is positive definite and $(X^T X)^{-1} = O(\alpha_n^{-1})$.*

Remark. Assuming $X^T X$ is nonsingular, for a LS estimator $\hat{\boldsymbol{\beta}}_{ols}$ to converge to true value $\boldsymbol{\beta}_0$ in probability, it is sufficient and necessary that $(X^T X)^{-1} \to 0$ as $n \to \infty$ (Anderson and Taylor, 1976; Lai et al., 1978).

Remark. Although we have stated this as an assumption, one typically assumes an n-dependence for α_n (Anderson and Taylor, 1976). Since the form of the n-dependence is unspecified, we can alternatively view Assumption 1 as a definition of α_n. The usual assumption that is made (typically for analytical convenience) is that $\alpha_n = n$ (Fu and Knight, 2000). We will provide examples of toy data for which $\alpha_n = n$, as well as examples for which $\alpha_n \neq n$. In light of our empirical results in Section 4 and the empirical observation that leverage scores are often very nonuniform (Mahoney and Drineas, 2009; Gittens and Mahoney, 2013), it is an interesting question to ask whether the common assumption that $\alpha_n = n$ is too restrictive, e.g., whether it excludes interesting matrices X with very heterogeneous leveraging scores.

Under Assumption 1, i.e., that $(X^T X)^{-1}$ is asymptotically parameterized as $(X^T X)^{-1} = O(\alpha_n^{-1})$, we have the following three results to compare the leveraging estimators and the uniform sampling estimator. The expressions in these three lemmas are complicated; and, since they are expressed in terms of α_n, they are not easy to evaluate on real or synthetic data. (It is partly for this reason that our empirical evaluation is in terms of the bias and variance of the subsampling estimators.) We start by stating a lemma characterizing the relative efficiency of LEV and UNIF; the proof of this lemma may be found in Appendix B.

Lemma 8 *To leading order, the asymptotic relative efficiency of $c^T \tilde{\boldsymbol{\beta}}_{LEV}$ and $c^T \tilde{\boldsymbol{\beta}}_{UNIF}$ is*

$$e(c^T \tilde{\boldsymbol{\beta}}_{LEV}, c^T \tilde{\boldsymbol{\beta}}_{UNIF}) \simeq O(\frac{\frac{1}{\alpha_n} + \frac{1}{r}\sqrt{\sum_i (1-h_{ii})^4 \max(h_{ii})}}{\frac{1}{\alpha_n} + \frac{1}{\alpha_n r}\sqrt{\sum_i \frac{(1-h_{ii})^4}{h_{ii}^2} \max(h_{ii})}}), \tag{24}$$

where the residual variance is ignored.

Next, we state a lemma characterizing the relative efficiency of SLEV and UNIF; the proof of this lemma is similar to that of Lemma 8 and is thus omitted.

Lemma 9 *To leading order, the asymptotic relative efficiency of $c^T \tilde{\boldsymbol{\beta}}_{SLEV}$ and $c^T \tilde{\boldsymbol{\beta}}_{UNIF}$ is*

$$e(c^T \tilde{\boldsymbol{\beta}}_{SLEV}, c^T \tilde{\boldsymbol{\beta}}_{UNIF}) \simeq O(\frac{\frac{1}{\alpha_n} + \frac{1}{r}\sqrt{\sum_i (1-h_{ii})^4 \max(h_{ii})}}{\frac{1}{\alpha_n} + \frac{1}{\alpha_n r}\sqrt{\sum_i \frac{(1-h_{ii})^4}{\pi_i^2} \max(h_{ii})}}),$$

where the residual variance is ignored.

Finally, we state a lemma characterizing the relative efficiency of LEVUNW and UNIF; the proof of this lemma may be found in Appendix B.

Lemma 10 *To leading order, the asymptotic relative efficiency of* $c^T\tilde{\boldsymbol{\beta}}_{LEVUNW}$ *and* $c^T\tilde{\boldsymbol{\beta}}_{UNIF}$ *is*

$$e(c^T\tilde{\boldsymbol{\beta}}_{LEVUNW}, c^T\tilde{\boldsymbol{\beta}}_{UNIF}) \simeq O(\frac{\frac{1}{\alpha_n} + \frac{1}{r}\sqrt{\sum_i (1-h_{ii})^4 \max(h_{ii})}}{\frac{\max(h_{ii})}{\alpha_n \min(h_{ii})} + \frac{1}{\alpha_n \min(h_{ii}) r}\sqrt{\sum_i (1-g_{ii})^4 \max(g_{ii})}}),$$

where the residual variance is ignored and $g_{ii} = h_{ii}\boldsymbol{x}_i^T(X^T Diag\{h_{ii}\}X)^{-1}\boldsymbol{x}_i$.

Of course, in an analogous manner, one could derive expressions for the asymptotic relative efficiencies $e(c^T\tilde{\boldsymbol{\beta}}_{SLEV}, c^T\tilde{\boldsymbol{\beta}}_{LEV})$, $e(c^T\tilde{\boldsymbol{\beta}}_{LEVUNW}, c^T\tilde{\boldsymbol{\beta}}_{LEV})$, and $e(c^T\tilde{\boldsymbol{\beta}}_{LEVUNW}, c^T\tilde{\boldsymbol{\beta}}_{SLEV})$.

A.2 Illustration of the Method on Toy Data

Here, we will consider several toy data sets that illustrate various aspects of algorithmic leveraging, including various extreme cases of the method. While some of these toy data may seem artificial or contrived, they will highlight properties that manifest themselves in less extreme forms in the more realistic data in Section 4. Since the leverage score structure of the matrix X is crucial for the behavior of the method, we will focus primarily on that structure. To do so, consider the two extreme cases. At one extreme, when the leverage scores are all equal, i.e., $h_{ii} = p/n$, for all $i \in [n]$, the first two variance terms in Eqn. (20) are equal to the first two variance terms in Eqn. (18). In this case, LEV simply reduces to UNIF. At the other extreme, the leverage scores can be very nonuniform—e.g., there can be a small number of leverage scores that are much larger than the rest and/or there can be some leverage scores that are much smaller than the mean score. Dealing with these two cases properly is crucial for the method of algorithmic leveraging, but these two cases highlight important differences between the more common algorithmic perspective and our more novel statistical perspective.

The former problem (of a small number of very large leverage scores) is of particular importance from an algorithmic perspective. The reason is that in that case one wants to compare the output of the sampling algorithm with the optimum based on the empirical data (as opposed to the "ground truth" solution). Thus, dealing with large leverage scores was a main issue in the development of the leveraging paradigm (Drineas et al., 2006; Mahoney, 2011; Drineas et al., 2012). On the other hand, the latter problem (of some very small leverage scores) is also an important concern if we are interested in statistical properties of algorithmic leveraging. To see why, consider, e.g., the extreme case that a few data points have very very small leverage scores, e.g. $h_{ii} = 1/n^4$ for some i. In this case, e.g., the second variance term in Eqn. (18) will be much larger than the second variance term in Eqn. (20).

In light of this discussion, here are several toy examples to consider. We will start with several examples where $p = 1$ that illustrate things in the simplest setting.

- **Example 1A: Sample Mean.** Let n be arbitrary, $p = 1$, and let the $n \times p$ matrix X be such that $X_i = 1$, for all $i \in [n]$, i.e., let X be the all-ones vector. In this case, $X^T X = n$ and $h_{ii} = 1/n$, for all $i \in [n]$, i.e., the leverage scores are uniform, and thus algorithmic leveraging reduces to uniform sampling. Also, in this case, $\alpha_n = n$ in Assumption 1. All three asymptotic efficiencies are equal to $O(1)$.

- **Example 1B: Simple Linear Combination.** Let n be arbitrary, $p = 1$, and let the $n \times p$ matrix X be such that $X_i = \pm 1$, for all $i \in [n]$, either uniformly at random, or such that $X_i = +1$ if i is odd and $X_i = -1$ if i is even. In this case, $X^T X = n$ and $h_{ii} = 1/n$, for all $i \in [n]$, i.e., the leverage scores are uniform; and, in addition, $\alpha_n = n$ in Assumption 1. For all four estimators, all four unconditional variances are equal to $\sigma^2\{\frac{1}{n} + \frac{(1-1/n)^2}{r}\}$. In addition, for all four estimators, all three relative efficiencies are equal to $O(1)$.

- **Example 2: "Domain Expansion" Regression Line Through Origin.** Let n be arbitrary, $p = 1$, and let the $n \times p$ matrix X be such that $X_i = i$, i.e., they are evenly spaced and increase without limit with increasing i. In this case,

$$X^T X = n(n+1)(2n+1)/6,$$

and the leverage scores equal

$$h_{ii} = \frac{6i^2}{n(n+1)(2n+1)},$$

i.e., the leverage scores h_{ii} are very nonuniform. This is illustrated in the left panel of Figure 16. Also, in this case, $\alpha_n = n^3$ in Assumption 1. It is easy to see that the first variance components of UNIF, LEV, SLEV are the same, i.e., they equal

$$(X^T X)^{-1} = \frac{6}{n(n+1)(2n+1)}.$$

It is also easy to see that variances of LEV, SLEV and UNIF are dominated by their second variance component. The leading terms of the second variance component of LEV and UNIF are the same, and we expect to see the similar performance based on their variance. The leading term of the second variance component of SLEV is smaller than that of LEV and UNIF; and thus SLEV has smaller variance than LEV and UNIF. Simple calculation shows that LEVUNW has a smaller leading term for the second variance component than those of LEV, UNIF and SLEV.

- **Example 3: "In-fill" Regression Line Through Origin.** Let n be arbitrary, $p = 1$, and let the $n \times p$ matrix X be such that $X_i = 1/i$. This is different than the evenly spaced data points in the "inflated" toy example since the unevenly spaced data points this this example get denser in the interval $(0, 1]$. The asymptotic properties of such design matrix are so-called "in-fill" asymptotics (Cressie, 1991). In this case,

$$X^T X = \pi^2/6 - \psi^{(1)}(n+1),$$

where $\psi^{(k)}$ is the k^{th} derivative of digamma function, and the leverage scores equal

$$h_{ii} = \frac{1}{i^2(\pi^2/6 - \psi^{(1)}(n+1))},$$

i.e., the leverage scores h_{ii} are very nonuniform. This is illustrated in the middle panel of Figure 16. Also, in this case, $\alpha_n = 1$ in Assumption 1.

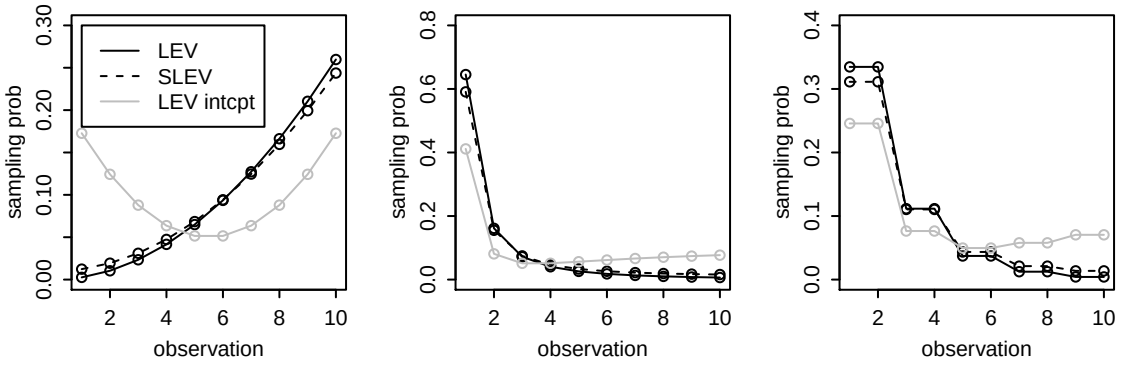


Figure 16: Leverage score-based sampling probabilities for three toy examples (Example 2, Example 3, and Example 4). Left panel is Inflated Regression Line (Example 2); Middle panel is In-fill Regression Line (Example 3); Right panel is Regression Surface (Example 4). In this example, we set $n = 10$. Black lines connect the sampling probability for each data points for LEV; dash lines (below black) connect sampling probability for SLEV; and grey lines connect sampling probability for LEV after we add an intercept (i.e., the sample mean) as a second column to X.

To obtain an improved understanding of these examples, consider the first two panels of Figures 16 and 17. Figure 16 shows the sampling probabilities for the Inflated Regression Line and the In-fill Regression Line. Both the Inflated Regression Line and the In-fill Regression Line have very nonuniform leverage scores, and by construction there is a natural ordering such that the leverage scores increase or decrease respectively. For the Inflated Regression Line, the minimum, mean, and maximum leverage scores are $6/(n(n+1)(2n+1))$, $1/n$, and $6n/(n+1)(2n+1)$, respectively; and for the In-fill Regression Line, the minimum, mean, and maximum leverage scores are $1/(n^2(\pi^2/6 - \psi^{(1)}(n+1)))$, $1/n$, and $1/(\pi^2/6 - \psi^{(1)}(n+1))$, respectively. For reference, note that for the Sample Mean (as well as for the Simple Linear Combination) all of the the leverage scores are equal to $1/n$, which equals 0.1 for the value of $n = 10$ used in Figure 16.

Figure 17 illustrates the theoretical variances for the same examples for particular values of σ^2 and r. In particular, observe that for the Inflated Regression Line, all three sampling methods tend to have smaller variance as n is increased for a fixed value of p. This is intuitive, and it is a common phenomenon that we observe in most of the synthetic and real data sets. The property of the In-fill Regression Line where the variances are roughly flat (actually, they increase slightly) is more uncommon, but it illustrates that other possibilities exist. The reason is that leverage scores of most data points are relatively homogeneous (as long as i is greater than $\sqrt{6n/\pi^2}$, the leverage score of ith observation is less than mean $1/n$ but greater than $1/n^2(\pi^2/6)$). When subsample size r is reasonably large, we have high probabilities to sample these data points, whose sample probabilities inflate the variance. These curves also illustrate that LEV and UNIF can be better or worse with respect to each

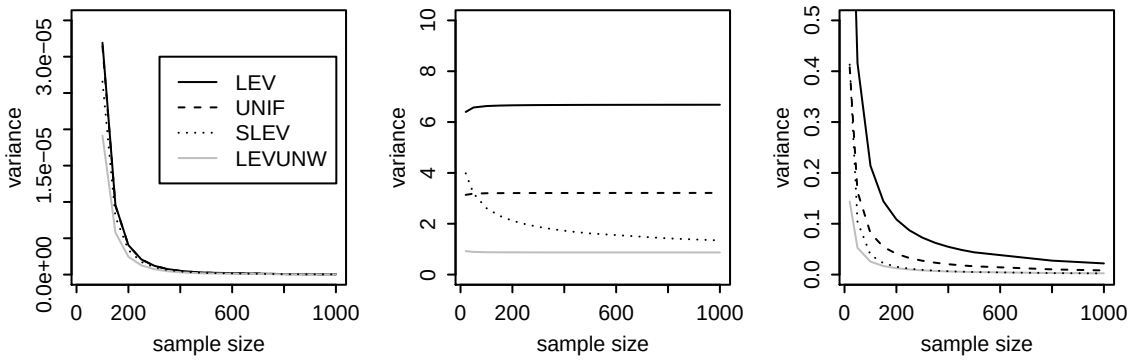


Figure 17: Theoretical variances for three toy examples (Example 2, Example 3, and Example 4) for various sample sizes n. Left panel is Inflated Regression Line (Example 2); Middle panel is In-fill Regression Line (Example 3); Right panel is Regression Surface (Example 4). In this example, we set $\sigma^2 = 1$ and $r = 0.1n$, for varying n from 100 to 1000. Black line for LEV (Equation 18); dash line for UNIF (Equation 20); dotted line (below black) for SLEV; and grey line for LEVUNW (Equation 23).

other, depending on the problem parameters; and that SLEV and LEVUNW can be better than either, for certain parameter values.

From these examples, we can see that the variance for the leveraging estimate can be inflated by very small leverage scores. That is, since the variances involve terms that depend on the inverse of h_{ii}, they can be large if h_{ii} is very small. Here, we note that the common practice of adding an intercept, i.e., a sample mean or all-ones vector *tends* to uniformize the leverage scores. That is, in statistical model building applications, we usually have intercept—which is an all-ones vector, called the Sample Mean above—in the model, i.e., the first column of X is $\mathbf{1}$ vector; and, in this case, the h_{ii}s are bounded below by $1/n$ and above by $1/w_i$ (Weisberg, 2005). This is also illustrated in Figure 16, which shows the the leverage scores for when an intercept is included. Interestingly, for the Inflated Regression Line, the scores for elements that originally had very small score actually increase to be on par with the largest scores. In our experience, it is much more common for the small leverage scores to simply be increased a bit, as is illustrated with the modified scores for the In-fill Regression Line.

We continue the toy examples with an example for $p = 2$; this is the simplest case that allows us to look at what is behind Assumption 1.

- **Example 4: Regression Surface Through Origin.** Let $p = 2$ and $n = 2k$ be even. Let the elements of X be defined as $\boldsymbol{x}_{2j-1,n} = \begin{pmatrix} \sqrt{\frac{n}{3^j}} & 0 \end{pmatrix}$, and $\boldsymbol{x}_{2j,n} = \begin{pmatrix} 0 & \sqrt{\frac{n}{3^j}} \end{pmatrix}$. In this case,

$$X^T X = (n \sum_{j=1}^{n} \frac{1}{3^j}) I_2 = k \frac{3^k - 1}{3^k} I_2 = O(n),$$

and the leverage scores equal

$$h_{2j-1,2j-1} = h_{2j,2j} = \frac{2 \times 3^k}{3^j(3^k - 1)}.$$

Here, $\alpha_n = n$ in Assumption 1, and the largest leverage score does *not* converge to zero.

To see the leverage scores and the (theoretically-determined) variance for the Regression Surface of Example 4, see the third panel of Figures 16 and 17. In particular, the third panel of Figure 16 demonstrates what we saw with the $p = 1$ examples, i.e., that adding an intercept tends to increase the small leverage scores; and Figure 17 illustrates that the variances of all four estimates are getting close as sample size n becomes larger.

Remark. It is worth noting that (Miller, 1974a) showed $\alpha_n = n$ in Assumption 1 implies that $\max h_{ii} \to 0$. In his proof, Miller essentially assumed that $\boldsymbol{x}_i$, $i = 1, \ldots, n$ is a single sequence. Example 4 shows that Miller's theorem does not hold for triangular array (with one pattern for even numbered observations and the other pattern for odd numbered observations) (Shao, 1987).

Finally, we consider several toy data sets with larger values of p. In this case, there starts to be a nontrivial interaction between the singular value structure and the singular vector structure of the matrix X.

- **Example 5: Truncated Hadamard Matrix.** An $n \times p$ matrix consisting of p columns from a Hadamard Matrix (which is an orthogonal matrix) has uniform leverage scores—all are equal. Similarly, for an $n \times p$ matrix with entries i.i.d. from Gaussian distribution—that is, unless the aspect ratio of the matrix is extremely rectangular, e.g., $p = 1$, the leverage scores of a random Gaussian matrix are very close to uniform. (In particular, as our empirical results demonstrate, using nonuniform sampling probabilities is not necessary for data generated from Gaussian random matrices.)

- **Example 6: Truncated Identity Matrix.** An $n \times p$ matrix consisting of the first p columns from an Identity Matrix (which is an orthogonal matrix) has very nonuniform leverage scores—the first p are large, and the remainder are zero. (Since one could presumably remove the all-zeros rows, this example might seem trivial, but it is useful as a worst-case thought experiment.)

- **Example 7: Worst-case Matrix.** An $n \times p$ matrix consisting of $n - 1$ rows all pointing in the same direction and 1 row pointing in some other direction. This has one leverage score—the one corresponding to the row pointing in the other direction—that is large, and the rest are mediumly-small. (This is an even better worst-case matrix than Example 6; and in the main text we have an even less trivial example of this.)

Example 5 is "nice" from an algorithmic perspective and, as seen in Section 4, from a statistical perspective as well. Since they have nonuniform leverage scores; Example 6 and Example 7 are worse from an algorithmic perspective. As our empirical results will demonstrate, they are also problematic from a statistical perspective, but for slightly different reasons.

Appendix B. Proofs of our main results

In this appendix, we will provide proofs of several of our main results.

B.1 Proof of Lemma 1

Recall that the matrix $W = S_X D^2 S_X^T$ encodes information about the sampling/rescaling process; in particular, this includes UNIF, LEV, and SLEV, although our results hold more generally.

By performing a Taylor expansion of $\tilde{\boldsymbol{\beta}}_W(\boldsymbol{w})$ around the point $\boldsymbol{w}_0 = \mathbf{1}$, we have

$$\tilde{\boldsymbol{\beta}}_W(\boldsymbol{w}) = \tilde{\boldsymbol{\beta}}_W(\boldsymbol{w}_0) + \frac{\partial \tilde{\boldsymbol{\beta}}_W(\boldsymbol{w})}{\partial \boldsymbol{w}^T}|_{\boldsymbol{w}=\boldsymbol{w}_0}(\boldsymbol{w} - \boldsymbol{w}_0) + R_W,$$

where R_W is remainder. Remainder $R_W = o_p(||\boldsymbol{w} - \boldsymbol{w}_0||)$ when $\boldsymbol{w}$ is close to $\boldsymbol{w}_0$. By setting $\boldsymbol{w}_0$ as the all-one vector, i.e., $\boldsymbol{w}_0 = \mathbf{1}$, $\tilde{\boldsymbol{\beta}}_W(\boldsymbol{w}_0)$ is expanded around the full sample ordinary LS estimate $\hat{\boldsymbol{\beta}}_{ols}$, i.e., $\tilde{\boldsymbol{\beta}}_W(\mathbf{1}) = \hat{\boldsymbol{\beta}}_{ols}$. That is,

$$\tilde{\boldsymbol{\beta}}_W(\boldsymbol{w}) = \hat{\boldsymbol{\beta}}_{ols} + \frac{\partial (X^T Diag\{\boldsymbol{w}\} X)^{-1} X^T Diag\{\boldsymbol{w}\} \boldsymbol{y}}{\partial \boldsymbol{w}^T}|_{\boldsymbol{w}=\mathbf{1}}(\boldsymbol{w} - \mathbf{1}) + R_W.$$

By differentiation by parts, we obtain

$$\begin{aligned}
\frac{\partial (X^T Diag\{\boldsymbol{w}\} X)^{-1} X^T Diag\{\boldsymbol{w}\} \boldsymbol{y}}{\partial \boldsymbol{w}^T} &= \frac{\partial \mathrm{Vec}[(X^T Diag\{\boldsymbol{w}\} X)^{-1} X^T Diag\{\boldsymbol{w}\} \boldsymbol{y}]}{\partial \boldsymbol{w}^T} \\
&= (\mathbf{1} \otimes (X^T Diag\{\boldsymbol{w}\} X)^{-1}) \frac{\partial \mathrm{Vec}[X^T Diag\{\boldsymbol{w}\} \boldsymbol{y}]}{\partial \boldsymbol{w}^T} \qquad (25) \\
&\quad + (\boldsymbol{y}^T Diag\{\boldsymbol{w}\} X \otimes I_p) \frac{\partial \mathrm{Vec}[(X^T Diag\{\boldsymbol{w}\} X)^{-1}]}{\partial \boldsymbol{w}^T} \qquad (26)
\end{aligned}$$

where Vec is Vec operator, which stacks the columns of a matrix into a vector, and $\otimes$ is the Kronecker product. The Kronecker product is defined as follows: suppose $A = \{a_{ij}\}$ is an $m \times n$ matrix and $B = \{b_{ij}\}$ is a $p \times q$ matrix; then, $A \otimes B$ is a $mp \times nq$ matrix, comprising m rows and n columns of $p \times q$ blocks, the ijth of which is $a_{ij}B$.

To simplify (25), note that is easy to show that (25) can be seen as

$$(\mathbf{1} \otimes (X^T Diag\{\boldsymbol{w}\} X)^{-1})(\boldsymbol{y}^T \otimes X^T) \frac{\partial \mathrm{Vec}[Diag\{\boldsymbol{w}\}]}{\partial \boldsymbol{w}^T}. \qquad (27)$$

To simplify (26), we need the following two results of matrix differentiation,

$$\begin{aligned}
\frac{\partial \mathrm{Vec}[X^{-1}]}{\partial (\mathrm{Vec}X)^T} &= -(X^{-1})^T \otimes X^{-1}, \text{ and} \\
\frac{\partial \mathrm{Vec}[AWB]}{\partial \boldsymbol{w}^T} &= (B^T \otimes A) \frac{\partial \mathrm{Vec}[W]}{\partial \boldsymbol{w}^T}, \qquad (28)
\end{aligned}$$

where the details on these two results can be found on page 366-367 of (Harville, 1997). By combining the two results in (28), by the chain rule, we have

$$\begin{aligned}
&\frac{\partial \text{Vec}[(X^T Diag\{\boldsymbol{w}\} X)^{-1}]}{\partial \boldsymbol{w}^T} \\
&= \frac{\partial \text{Vec}[(X^T Diag\{\boldsymbol{w}\} X)^{-1}]}{\partial \text{Vec}[(X^T Diag\{\boldsymbol{w}\} X)]^T} \frac{\partial \text{Vec}[(X^T Diag\{\boldsymbol{w}\} X)]}{\partial \boldsymbol{w}^T} \\
&= -(X^T Diag\{\boldsymbol{w}\} X)^{-1} \otimes (X^T Diag\{\boldsymbol{w}\} X)^{-1}(X^T \otimes X^T) \frac{\partial \text{Vec}[Diag\{\boldsymbol{w}\}]}{\partial \boldsymbol{w}^T}
\end{aligned}$$

By simple but tedious algebra, (25) and (26) give rise to

$$\begin{aligned}
\{(\boldsymbol{y}^T - \boldsymbol{y}^T Diag\{\boldsymbol{w}\} X(X^T Diag\{\boldsymbol{w}\} X)^{-1} X^T) \otimes (X^T Diag\{\boldsymbol{w}\} X)^{-1} X^T\} \frac{\partial \text{Vec}[Diag\{\boldsymbol{w}\}]}{\partial \boldsymbol{w}^T} \\
= \{(\boldsymbol{y} - X\tilde{\boldsymbol{\beta}}_W(\boldsymbol{w}))^T \otimes (X^T Diag\{\boldsymbol{w}\} X)^{-1} X^T\} \frac{\partial \text{Vec}[Diag\{\boldsymbol{w}\}]}{\partial \boldsymbol{w}^T} \quad (29)
\end{aligned}$$

By combining these results, we thus have,

$$\begin{aligned}
\tilde{\boldsymbol{\beta}}_W &= \hat{\boldsymbol{\beta}}_{ols} + \{(\boldsymbol{y} - X\hat{\boldsymbol{\beta}}_{ols})^T \otimes (X^T X)^{-1} X^T\} \frac{\partial \text{Vec}(Diag\{\boldsymbol{w}\})}{\partial \boldsymbol{w}^T} (\boldsymbol{w} - \mathbf{1}) + R_W \\
&= \hat{\boldsymbol{\beta}}_{ols} + \{\hat{\boldsymbol{e}}^T \otimes (X^T X)^{-1} X^T\} \begin{pmatrix} \mathbf{e}_1 \mathbf{e}_1^T \\ \mathbf{e}_2 \mathbf{e}_2^T \\ \\ \\ \mathbf{e}_n \mathbf{e}_n^T \end{pmatrix} (\boldsymbol{w} - \mathbf{1}) + R_W \\
&= \hat{\boldsymbol{\beta}}_{ols} + (X^T X)^{-1} X^T Diag\{\hat{\boldsymbol{e}}\} (\boldsymbol{w} - \mathbf{1}) + R_W
\end{aligned}$$

where $\hat{\boldsymbol{e}} = \boldsymbol{y} - X\hat{\boldsymbol{\beta}}_{ols}$ is the LS residual vector, $\mathbf{e}_i$ is a length n vector with i^{th} element equal to one and all other elements equal to zero, from which the lemma follows.

B.2 Proof of Lemma 2

Recall that we will use W to refer to the sampling process.

We start by establishing the conditional result. Since $\mathbf{E}[\boldsymbol{w}] = \mathbf{1}$, it is straightforward to calculate conditional expectation of $\tilde{\boldsymbol{\beta}}_W$. Then, it is easy to see that

$$\begin{aligned}
\mathbf{E}[(w_i - 1)(w_j - 1)] &= \frac{1}{r\pi_i} - \frac{1}{r} \quad \text{for} \quad i = j \\
&= -\frac{1}{r} \quad \text{for} \quad i \neq j.
\end{aligned}$$

We rewrite it in matrix form,

$$\mathbf{Var}[\boldsymbol{w}] = \mathbf{E}\left[(\boldsymbol{w} - \mathbf{1})(\boldsymbol{w} - \mathbf{1})^T\right] = Diag\left\{\frac{1}{r\boldsymbol{\pi}}\right\} - \frac{1}{r} J_n,$$

where $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_n)^T$ and J_n is a $n \times n$ matrix of ones. Some additional algebra yields that the variance of $\tilde{\boldsymbol{\beta}}_W$ is

$$\begin{aligned}
\mathbf{Var_w}\left[\tilde{\boldsymbol{\beta}}_W - \hat{\boldsymbol{\beta}}|\boldsymbol{y}\right] &= \mathbf{Var}\left[(X^TX)^{-1}X^T Diag\left\{\hat{\boldsymbol{e}}\right\}(\boldsymbol{w}-\mathbf{1})|\boldsymbol{y}\right] + \mathbf{Var_w}\left[R_W\right] \\
&= (X^TX)^{-1}X^T Diag\left\{\hat{\boldsymbol{e}}\right\}(Diag\left\{\frac{1}{r\boldsymbol{\pi}}\right\} - \frac{1}{r}J_n) Diag\left\{\hat{\boldsymbol{e}}\right\} X(X^TX)^{-1} \\
&+ \mathbf{Var_w}\left[R_W\right] \\
&= (X^TX)^{-1}X^T[Diag\left\{\hat{\boldsymbol{e}}\right\} Diag\left\{\frac{1}{r\boldsymbol{\pi}}\right\} Diag\left\{\hat{\boldsymbol{e}}\right\}]X(X^TX)^{-1} + \mathbf{Var}\left[R_W\right] \\
&= (X^TX)^{-1}X^T Diag\left\{\frac{1}{r\boldsymbol{\pi}}\hat{\boldsymbol{e}}^2\right\} X(X^TX)^{-1} + \mathbf{Var_w}\left[R_W\right].
\end{aligned}$$

Setting $\pi_i = h_{ii}/p$ in above equations, we thus prove the conditional result.

We next establish the unconditional result as follows. The unconditional expectation result is easy to see as each data point is unbiased to $\boldsymbol{\beta}_0$. By rule of double expectations, we have the variance of $\tilde{\boldsymbol{\beta}}_W$ result, from which the lemma follows.

B.3 Proof of Lemma 5

First note that the unweighted leveraging estimate $\tilde{\boldsymbol{\beta}}_{LEVUNW}$ can be written as

$$\tilde{\boldsymbol{\beta}}_{LEVUNW} = (X^TS_XS_X^TX)^{-1}X^TS_XS_X^T\boldsymbol{y} = (X^TW_{LEVUNW}X)^{-1}X^TW_{LEVUNW}\boldsymbol{y},$$

where $W_{LEVUNW} = S_XS_X^T = Diag\left\{\boldsymbol{w}_{LEVUNW}\right\}$, and where $\boldsymbol{w}_{LEVUNW}$ has a multinomial distribution $Multi(r, \boldsymbol{\pi})$. The proof of this lemma is analogous to the proof of Lemma 1; and so here we provide only some details on the differences. By employing a Taylor expansion, we have

$$\begin{aligned}
\tilde{\boldsymbol{\beta}}_{LEVUNW}(\boldsymbol{w}_{LEVUNW}) = \tilde{\boldsymbol{\beta}}_{LEVUNW}(\boldsymbol{w}_0) + \frac{\partial\tilde{\boldsymbol{\beta}}_{LEVUNW}(\boldsymbol{w})}{\partial\boldsymbol{w}^T}|_{\boldsymbol{w}=\boldsymbol{w}_0}(\boldsymbol{w}_{LEVUNW} - \boldsymbol{w}_0) \\
+ R_{LEVUNW},
\end{aligned}$$

where $R_{LEVUNW} = o_p(||\boldsymbol{w}_{LEVUNW} - \boldsymbol{w}_0||)$. Following the proof of the previous lemma, we have that

$$\begin{aligned}
\tilde{\boldsymbol{\beta}}_{LEVUNW} &= \hat{\boldsymbol{\beta}}_{wls} + \{(\boldsymbol{y} - X\hat{\boldsymbol{\beta}}_{wls})^T \otimes (X^TW_0X)^{-1}X^T\}\frac{\partial \mathrm{vec}(Diag\left\{\boldsymbol{w}_{LEVUNW}\right\})}{\partial\boldsymbol{w}_{LEVUNW}^T} \\
&\quad (\boldsymbol{w}_{LEVUNW} - \boldsymbol{w}_0) + R_{LEVUNW} \\
&= \hat{\boldsymbol{\beta}}_{wls} + \{\hat{\boldsymbol{e}}_w^T \otimes (X^TW_0X)^{-1}X^T\}\begin{pmatrix} \mathbf{e}_1\mathbf{e}_1^T & & & \\ & \mathbf{e}_2\mathbf{e}_2^T & & \\ & & & \\ & & & \mathbf{e}_n\mathbf{e}_n^T \end{pmatrix}(\boldsymbol{w}_{LEVUNW} - \boldsymbol{w}_0) + R_{LEVUNW} \\
&= \hat{\boldsymbol{\beta}}_{wls} + (X^TW_0X)^{-1}X^T Diag\left\{\hat{\boldsymbol{e}}_{w_0}\right\}(\boldsymbol{w}_{LEVUNW} - \boldsymbol{w}_0) + R_{LEVUNW},
\end{aligned}$$

where $W_0 = Diag\left\{\boldsymbol{w}_0\right\} = Diag\left\{r\boldsymbol{\pi}\right\}$, $\hat{\boldsymbol{\beta}}_{wls} = (X^TW_0X)^{-1}X^TW_0\boldsymbol{y}$, $\hat{\boldsymbol{e}}_w = \boldsymbol{y} - X\hat{\boldsymbol{\beta}}_{wls}$ is the weighted LS residual vector, $\mathbf{e}_i$ is a length n vector with i^{th} element equal to one and all other elements equal to zero. From this the lemma follows.

B.4 Proof of Lemma 6

By taking the conditional expectation of Taylor expansion of the LEVUNW estimate $\tilde{\boldsymbol{\beta}}_{LEVUNW}$ in Lemma 5, we have that

$$\mathbf{E_w}\left[\tilde{\boldsymbol{\beta}}_{LEVUNW}|\boldsymbol{y}\right] = \hat{\boldsymbol{\beta}}_{wls} + (X^TW_0X)^{-1}X^T Diag\left\{\hat{\boldsymbol{e}}_w\right\}\mathbf{E_w}\left[\boldsymbol{w} - r\boldsymbol{\pi}\right] + \mathbf{E_w}\left[R_{LEVUNW}\right].$$

Since $\mathbf{E_w}\left[\boldsymbol{w}_{LEVUNW}\right] = r\boldsymbol{\pi}$, the conditional expectation is thus obtained. Since $\boldsymbol{w}_{LEVUNW}$ is multinomial distributed, we have

$$\mathbf{Var}\left[\boldsymbol{w}_{LEVUNW}\right] = \mathbf{E}\left[(\boldsymbol{w}_{LEVUNW} - r\boldsymbol{\pi})(\boldsymbol{w}_{LEVUNW} - r\boldsymbol{\pi})^T\right] = Diag\left\{r\boldsymbol{\pi}\right\} - r\boldsymbol{\pi}\boldsymbol{\pi}^T.$$

Some algebra yields that the conditional variance of $\tilde{\boldsymbol{\beta}}_{LEVUNW}$ is

$$\begin{aligned}
&\mathbf{Var_w}\left[\tilde{\boldsymbol{\beta}}_{LEVUNW} - \hat{\boldsymbol{\beta}}_{wls}|\boldsymbol{y}\right] \\
&\quad= \mathbf{Var_w}\left[(X^TW_0X)^{-1}X^T Diag\left\{\hat{\boldsymbol{e}}_w\right\}(\boldsymbol{w}_{LEVUNW} - r\boldsymbol{\pi})|\boldsymbol{y}\right] + \mathbf{Var_w}\left[R_{LEVUNW}\right] \\
&\quad= (X^TW_0X)^{-1}X^T Diag\left\{\hat{\boldsymbol{e}}_w\right\}W_0 Diag\left\{\hat{\boldsymbol{e}}_w\right\}X(X^TW_0X)^{-1} + \mathbf{Var_w}\left[R_{LEVUNW}\right].
\end{aligned}$$

Finally, note that

$$\mathbf{E}\left[\hat{\boldsymbol{\beta}}_{wls}\right] = (X^TW_0X)^{-1}XW_0\mathbf{E}\left[\boldsymbol{y}\right] = (X^TW_0X)^{-1}XW_0X\boldsymbol{\beta}_0 = \boldsymbol{\beta}_0.$$

From this the lemma follows.

B.5 Proof of Lemma 8

Since $\mathrm{Var}(c^T\tilde{\boldsymbol{\beta}}_{LEV}) = c^T\mathrm{Var}(\tilde{\boldsymbol{\beta}}_{LEV})c$, we shall the derive the asymptotic order of $\mathrm{Var}(\tilde{\boldsymbol{\beta}}_{LEV})$. The second variance component of $\tilde{\boldsymbol{\beta}}_{LEV}$ in (18) is seen to be

$$\begin{aligned}
&\frac{p\sigma^2}{r}(X^TX)^{-1}X^T Diag\left\{\frac{(1-h_{ii})^2}{h_{ii}}\right\}X(X^TX)^{-1} \\
&\quad= \frac{p\sigma^2}{r}\sum_i \frac{(1-h_{ii})^2}{h_{ii}}(X^TX)^{-1}\boldsymbol{x}_i\boldsymbol{x}_i^T(X^TX)^{-1} \\
&\quad\leq \frac{p\sigma^2}{r}\sqrt{\sum_i \frac{(1-h_{ii})^4}{h_{ii}^2}\sum_i((X^TX)^{-1}\boldsymbol{x}_i\boldsymbol{x}_i^T(X^TX)^{-1})^2},
\end{aligned}$$

where Cauchy-Schwartz inequality has been used. Next, we show that

$$\sum_i((X^TX)^{-1}\boldsymbol{x}_i\boldsymbol{x}_i^T(X^TX)^{-1})^2 = O(\max(h_{ii})\alpha_n^{-2}).$$

To see this, observe that

$$\begin{aligned}
\sum_i((X^TX)^{-1}\boldsymbol{x}_i\boldsymbol{x}_i^T(X^TX)^{-1})^2 &\leq \max((X^TX)^{-1}\boldsymbol{x}_i\boldsymbol{x}_i^T)\sum_i(X^TX)^{-2}\boldsymbol{x}_i\boldsymbol{x}_i^T(X^TX)^{-1} \\
&\leq \max(\boldsymbol{x}_i^T(X^TX)^{-1}\boldsymbol{x}_i)\sum_i(X^TX)^{-2}\boldsymbol{x}_i\boldsymbol{x}_i^T(X^TX)^{-1} \\
&= \max(\boldsymbol{x}_i^T(X^TX)^{-1}\boldsymbol{x}_i)(X^TX)^{-2} \\
&= O(\max(h_{ii})\alpha_n^{-2})
\end{aligned}$$

Thus, the second variance component of $\tilde{\boldsymbol{\beta}}_{LEV}$ in (18) is of the order of

$$O(\frac{1}{\alpha_n r}\sqrt{\sum_i \frac{(1-h_{ii})^4}{h_{ii}^2}\max(h_{ii})}).$$

Analogously, the second variance component of $\tilde{\boldsymbol{\beta}}_{UNIF}$ in (20) is of the order of

$$O(\frac{1}{r}\sqrt{\sum_i (1-h_{ii})^4\max(h_{ii})}).$$

The lemma then follows immediately.

B.6 Proof of Lemma 10

It is easy to see that $(X^T Diag\{h_{ii}\}X)^{-1} = O(1/(\min(h_{ii})\alpha_n))$. The second variance component of $\tilde{\boldsymbol{\beta}}_{LEVUNW}$ in (23) is seen to be

$$\begin{aligned}
&\frac{p\sigma^2}{r}(X^T Diag\{h_{ii}\}X)^{-1}X^T Diag\left\{(1-g_{ii})^2 h_{ii}\right\}X(X^T Diag\{h_{ii}\}X)^{-1} \\
&\quad = \frac{p\sigma^2}{r}\sum_i (1-g_{ii})^2 h_{ii}(X^T Diag\{h_{ii}\}X)^{-1}\boldsymbol{x}_i\boldsymbol{x}_i^T(X^T Diag\{h_{ii}\}X)^{-1} \\
&\quad \le \frac{p\sigma^2}{r}\sqrt{\sum_i (1-g_{ii})^4 \sum_i (h_{ii}(X^T Diag\{h_{ii}\}X)^{-1}\boldsymbol{x}_i\boldsymbol{x}_i^T(X^T Diag\{h_{ii}\}X)^{-1})^2},
\end{aligned}$$

where Cauchy-Schwartz inequality has used. Next, we show that

$$\sum_i (h_{ii}(X^T Diag\{h_{ii}\}X)^{-1}\boldsymbol{x}_i\boldsymbol{x}_i^T(X^T Diag\{h_{ii}\}X)^{-1})^2 = O(\max(g_{ii})(\min(h_{ii})\alpha_n)^{-2}).$$

To see this, observe that

$$\begin{aligned}
&\sum_i (h_{ii}(X^T Diag\{h_{ii}\}X)^{-1}\boldsymbol{x}_i\boldsymbol{x}_i^T(X^T Diag\{h_{ii}\}X)^{-1})^2 \\
&\le \max(h_{ii}(X^T Diag\{h_{ii}\}X)^{-1}\boldsymbol{x}_i\boldsymbol{x}_i^T)\sum_i h_{ii}(X^T Diag\{h_{ii}\}X)^{-2}\boldsymbol{x}_i\boldsymbol{x}_i^T(X^T Diag\{h_{ii}\}X)^{-1} \\
&\le \max(h_{ii}\boldsymbol{x}_i^T(X^T Diag\{h_{ii}\}X)^{-1}\boldsymbol{x}_i)\sum_i h_{ii}(X^T Diag\{h_{ii}\}X)^{-2}\boldsymbol{x}_i\boldsymbol{x}_i^T(X^T Diag\{h_{ii}\}X)^{-1} \\
&= \max(h_{ii}\boldsymbol{x}_i^T(X^T Diag\{h_{ii}\}X)^{-1}\boldsymbol{x}_i)(X^T Diag\{h_{ii}\}X)^{-2} = O(\max(g_{ii})(\min(h_{ii})\alpha_n)^{-2}).
\end{aligned}$$

Thus, the second variance component of $\tilde{\boldsymbol{\beta}}_{LEVUNW}$ in (23) is of the order of

$$O(\frac{1}{\alpha_n \min(h_{ii})r}\sqrt{\sum_i (1-g_{ii})^4\max(g_{ii})}).$$

The lemma then follows immediately.

References

N. Ailon and B. Chazelle. Faster dimension reduction. *Communications of the ACM*, 53 (2):97–104, 2010.

T. W. Anderson and J. B. Taylor. Strong consistency of least squares estimates in normal linear regression. *Annals of Statistics*, 4(4):788–790, 1976.

H. Avron, P. Maymounkov, and S. Toledo. Blendenpik: Supercharging LAPACK's least-squares solver. *SIAM Journal on Scientific Computing*, 32:1217–1236, 2010.

P. J. Bickel, F. Gotze, and W. R. van Zwet. Resampling fewer than n observations: gains, losses, and remedies for losses. *Statistica Sinica*, 7:1–31, 1997.

S. Chatterjee and A. S. Hadi. Influential observations, high leverage points, and outliers in linear regression. *Statistical Science*, 1(3):379–393, 1986.

K. L. Clarkson and D. P. Woodruff. Low rank approximation and regression in input sparsity time. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing*, pages 81–90, 2013.

K. L. Clarkson, P. Drineas, M. Magdon-Ismail, M. W. Mahoney, X. Meng, and D. P. Woodruff. The Fast Cauchy Transform and faster robust linear regression. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 466–477, 2013.

N. Cloonan, A.R. Forrest, G. Kolle, B.B. Gardiner, G.J. Faulkner, M.K. Brown, D.F. Taylor, A.L. Steptoe, S. Wani, G. Bethel, A.J. Robertson, A.C. Perkins, S.J. Bruce, C.C. Lee, S.S. Ranade, H.E. Peckham, J.M. Manning, K.J. McKernan, and S.M. Grimmond. Stem cell transcriptome profiling via massive-scale mRNA sequencing. *Nature Methods*, 5(7): 613–619, 2008.

N. Cressie. *Statistics for Spatial Data*. Wiley, New York, 1991.

D. Dalpiaz, X. He, and P. Ma. Bias correction in RNA-Seq short-read counts using penalized regression. *Statistics in Biosciences*, 5(1):88–99, 2013.

P. Dhillon, Y. Lu, D. P. Foster, and L. Ungar. New subsampling algorithms for fast least squares regression. In *Advances in Neural Information Processing Systems*, pages 360–368, 2013.

P. Drineas, M. W. Mahoney, and S. Muthukrishnan. Sampling algorithms for ℓ_2 regression and applications. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1127–1136, 2006.

P. Drineas, M. W. Mahoney, S. Muthukrishnan, and T. Sarlós. Faster least squares approximation. *Numerische Mathematik*, 117(2):219–249, 2010.

P. Drineas, M. Magdon-Ismail, M. W. Mahoney, and D. P. Woodruff. Fast approximation of matrix coherence and statistical leverage. *Journal of Machine Learning Research*, 13: 3475–3506, 2012.

B. Efron. Bootstrap methods: another look at the jackknife. *Annals of Statistics*, 7(1): 1–26, 1979.

B. Efron and G. Gong. A leisurely look at the bootstrap, the jackknife, and cross-validation. *American Statistician*, 37(1):36–48, 1983.

W. Fu and K. Knight. Asymptotics for lasso-type estimators. *Annals of Statistics*, 28: 1356–1378, 2000.

A. Gittens and M. W. Mahoney. Revisiting the Nyström method for improved large-scale machine learning. Technical report, 2013. Preprint: arXiv:1303.1849.

G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, 1996.

D. A. Harville. *Matrix Algebra from a Statistician's Perspective*. Springer-Verlag, New York, 1997.

D. V. Hinkley. Jackknifing in unbalanced situations. *Technometrics*, 19(3):285–292, 1977.

D. C. Hoaglin and R. E. Welsch. The hat matrix in regression and ANOVA. *American Statistician*, 32(1):17–22, 1978.

D. Hsu, S. M. Kakade, and T. Zhang. Random design analysis of ridge regression. *Foundations of Computational Mathematics*, 14(3):569–600, 2014.

L. Jaeckel. The infinitesimal jackknife. *Bell Laboratories Memorandum*, MM:72–1215–11, 1972.

A. Kleiner, A. Talwalkar, P. Sarkar, and M. Jordan. The big data bootstrap. In *Proceedings of the 29th International Conference on Machine Learning*, 2012.

T. L. Lai, H. Robbins, and C. Z. Wei. Strong consistency of least squares estimates in multiple regression. *Proceedings of National Academy of Sciences*, 75(7):3034–3036, 1978.

J. Li, H. Jiang, and W. H. Wong. Modeling non-uniformity in short-read rates in RNA-seq data. *Genome Biology*, 11:R50, 2010.

J. S. Liu, R. Chen, and W. H. Wong. Rejection control and sequential importance sampling. *Journal of the American Statistical Association*, 93(443):1022–1031, 1998.

M. W. Mahoney. *Randomized Algorithms for Matrices and Data*. Foundations and Trends in Machine Learning. NOW Publishers, Boston, 2011. Also available at: arXiv:1104.5557.

M. W. Mahoney and P. Drineas. CUR matrix decompositions for improved data analysis. *Proceedings of National Academy of Sciences*, 106:697–702, 2009.

X. Meng and M. W. Mahoney. Low-distortion subspace embeddings in input-sparsity time and applications to robust linear regression. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing*, pages 91–100, 2013.

X. Meng, M. A. Saunders, and M. W. Mahoney. LSRN: A parallel iterative solver for strongly over- or under-determined systems. *SIAM Journal on Scientific Computing*, 36 (2):C95–C118, 2014.

R. G. Miller. An unbalanced jackknife. *Annals of Statistics*, 2(5):880–891, 1974a.

R. G. Miller. The jackknife–a review. *Biometrika*, 61(1):1–15, 1974b.

T. Nielsen, R. B. West, S. C. Linn, O. Alter, M. A. Knowling, J. O'Connell, S. Zhu, M. Fero, G. Sherlock, J. R. Pollack, P. O. Brown, D. Botstein, and M. van de Rijn. Molecular characterisation of soft tissue tumours: a gene expression study. *Lancet*, 359(9314):1301–1307, 2002.

P. Paschou, E. Ziv, E. G. Burchard, S. Choudhry, W. Rodriguez-Cintron, M. W. Mahoney, and P. Drineas. PCA-correlated SNPs for structure identification in worldwide human populations. *PLoS Genetics*, 3:1672–1686, 2007.

P. Paschou, J. Lewis, A. Javed, and P. Drineas. Ancestry informative markers for fine-scale individual assignment to worldwide populations. *Journal of Medical Genetics*, page doi:10.1136/jmg.2010.078212, 2010.

D. N. Politis, J. P. Romano, and M. Wolf. *Subsampling*. Springer-Verlag, New York, 1999.

D. B. Rubin. The Bayesian bootstrap. *Annals of Statistics*, 9(1):130–134, 1981.

R. Serfling. Asymptotic relative efficiency in estimation. In Miodrag Lovric, editor, *International Encyclopedia of Statistical Sciences*, pages 68–72. Springer, 2010.

J. Shao. *On Resampling Methods for Variance Estimation and Related Topics*. PhD thesis, University of Wisconsin at Madison, 1987.

J. Shao and D. Tu. *The Jackknife and Bootstrap*. Springer-Verlag, New York, 1995.

P. F. Velleman and R. E. Welsch. Efficient computing of regression diagnostics. *American Statistician*, 35(4):234–242, 1981.

S. Weisberg. *Applied Linear Regression*. Wiley, New York, 2005.

C. F. J. Wu. Jackknife, bootstrap and other resampling methods in regression analysis. *Annals of Statistics*, 14(4):1261–1295, 1986.

J. Yang, X. Meng, and M. W. Mahoney. Quantile regression for large-scale applications. Technical report, 2013. Preprint: arXiv:1305.0087.

Journal of Machine Learning Research 16 (2015) 913-960 Submitted 10/13; Revised 9/14; Published 4/15

Distributed Matrix Completion and Robust Factorization

Lester Mackey† LMACKEY@STANFORD.EDU
Stanford University
Department of Statistics
390 Serra Mall
Stanford, CA 94305

Ameet Talwalkar† ATALWALKAR@GMAIL.COM
University of California, Los Angeles
Computer Science Department
4732 Boelter Hall
Los Angeles, CA 90095

Michael I. Jordan JORDAN@CS.BERKELEY.EDU
University of California, Berkeley
Department of Electrical Engineering and Computer Science and Department of Statistics
465 Soda Hall
Berkeley, CA 94720

Editor: Nathan Srebro
† These authors contributed equally.

Abstract

If learning methods are to scale to the massive sizes of modern data sets, it is essential for the field of machine learning to embrace parallel and distributed computing. Inspired by the recent development of matrix factorization methods with rich theory but poor computational complexity and by the relative ease of mapping matrices onto distributed architectures, we introduce a scalable divide-and-conquer framework for noisy matrix factorization. We present a thorough theoretical analysis of this framework in which we characterize the statistical errors introduced by the "divide" step and control their magnitude in the "conquer" step, so that the overall algorithm enjoys high-probability estimation guarantees comparable to those of its base algorithm. We also present experiments in collaborative filtering and video background modeling that demonstrate the near-linear to superlinear speed-ups attainable with this approach.

Keywords: collaborative filtering, divide-and-conquer, matrix completion, matrix factorization, parallel and distributed algorithms, randomized algorithms, robust matrix factorization, video surveillance

1. Introduction

The scale of modern scientific and technological data sets poses major new challenges for computational and statistical science. Data analyses and learning algorithms suitable for modest-sized data sets are often entirely infeasible for the terabyte and petabyte data sets that are fast becoming the norm. There are two basic responses to this challenge. One response is to abandon algorithms that have superlinear complexity, focusing attention on simplified algorithms that—in the setting of massive data—may achieve satisfactory results

because of the statistical strength of the data. While this is a reasonable research strategy, it requires developing suites of algorithms of varying computational complexity for each inferential task and calibrating statistical and computational efficiencies. There are many open problems that need to be solved if such an effort is to bear fruit.

The other response to the massive data problem is to retain existing algorithms but to apply them to subsets of the data. To obtain useful results under this approach, one embraces parallel and distributed computing architectures, applying existing base algorithms to multiple subsets of the data in parallel and then combining the results. Such a divide-and-conquer methodology has two main virtues: (1) it builds directly on algorithms that have proven their value at smaller scales and that often have strong theoretical guarantees, and (2) it requires little in the way of new algorithmic development. The major challenge, however, is in preserving the theoretical guarantees of the base algorithm once one embeds the algorithm in a computationally-motivated divide-and-conquer procedure. Indeed, the theoretical guarantees often refer to subtle statistical properties of the data-generating mechanism (e.g., sparsity, information spread, and near low-rankedness). These may or may not be retained under the "divide" step of a putative divide-and-conquer solution. In fact, we generally would expect subsampling operations to damage the relevant statistical structures. Even if these properties are preserved, we face the difficulty of combining the intermediary results of the "divide" step into a final consilient solution to the original problem. The question, therefore, is whether we can design divide-and-conquer algorithms that manage the tradeoffs relating these statistical properties to the computational degrees of freedom such that the overall algorithm provides a scalable solution that retains the theoretical guarantees of the base algorithm.

In this paper,[1] we explore this issue in the context of an important class of machine learning algorithms—the matrix factorization algorithms underlying a wide variety of practical applications, including collaborative filtering for recommender systems , e.g., Koren et al. (2009) and the references therein, link prediction for social networks (Hoff, 2005), click prediction for web search (Das et al., 2007), video surveillance (Candès et al., 2011), graphical model selection (Chandrasekaran et al., 2009), document modeling (Min et al., 2010), and image alignment (Peng et al., 2010). We focus on two instances of the general matrix factorization problem: noisy matrix completion (Candès and Plan, 2010), where the goal is to recover a low-rank matrix from a small subset of noisy entries, and noisy robust matrix factorization (Candès et al., 2011; Chandrasekaran et al., 2009), where the aim is to recover a low-rank matrix from corruption by noise and outliers of arbitrary magnitude. These two classes of matrix factorization problems have attracted significant interest in the research community.

Various approaches have been proposed for scalable noisy matrix factorization problems, in particular for noisy matrix completion, though the vast majority tackle rank-constrained non-convex formulations of these problems with no assurance of finding optimal solutions (Zhou et al., 2008; Gemulla et al., 2011; Recht and Ré, 2011; F. Niu et al., 2011; Yu et al., 2012). In contrast, convex formulations of noisy matrix factorization relying on the nuclear norm have been shown to admit strong theoretical estimation guarantees (Agarwal et al., 2011; Candès et al., 2011; Candès and Plan, 2010; Negahban and Wainwright, 2012),

1. A preliminary form of this work appears in Mackey et al. (2011).

and a variety of algorithms (e.g., Lin et al., 2009b; Ma et al., 2011; Toh and Yun, 2010) have been developed for solving both matrix completion and robust matrix factorization via convex relaxation. Unfortunately, however, all of these methods are inherently sequential, and all rely on the repeated and costly computation of truncated singular value decompositions (SVDs), factors that severely limit the scalability of the algorithms. Moreover, previous attempts at reducing this computational burden have introduced approximations without theoretical justification (Mu et al., 2011).

To address this key problem of noisy matrix factorization in a scalable and theoretically sound manner, we propose a divide-and-conquer framework for large-scale matrix factorization. Our framework, entitled Divide-Factor-Combine (DFC), randomly divides the original matrix factorization task into cheaper subproblems, solves those subproblems in parallel using a base matrix factorization algorithm for nuclear norm regularized formulations, and combines the solutions to the subproblems using efficient techniques from randomized matrix approximation. We develop a thoroughgoing theoretical analysis for the DFC framework, linking statistical properties of the underlying matrix to computational choices in the algorithms and thereby providing conditions under which statistical estimation of the underlying matrix is possible. We also present experimental results for several DFC variants demonstrating that DFC can provide near-linear to superlinear speed-ups in practice. Indeed, DFC naturally handles massive data sets that are too large to fit on a single machine, as DFC's minimal communication footprint is particularly well-suited for distributed computing environments.

The remainder of the paper is organized as follows. In Section 2, we define the setting of noisy matrix factorization and introduce the components of the DFC framework. Secs. 3, 4, and 5 present our theoretical analysis of DFC, along with a new analysis of convex noisy matrix completion and a novel characterization of randomized matrix approximation algorithms. To illustrate the practical speed-up and robustness of DFC, we present experimental results on collaborative filtering, video background modeling, and simulated data in Section 6. Finally, we conclude in Section 7.

Notation: For a matrix $\mathbf{M} \in \mathbb{R}^{m \times n}$, we define $\mathbf{M}_{(i)}$ as the ith row vector, $\mathbf{M}^{(j)}$ as the jth column vector, and $\mathbf{M}_{ij}$ as the ijth entry. If $\operatorname{rank}(\mathbf{M}) = r$, we write the compact singular value decomposition (SVD) of $\mathbf{M}$ as $\mathbf{U}_M \mathbf{\Sigma}_M \mathbf{V}_M^\top$, where $\mathbf{\Sigma}_M$ is diagonal and contains the r non-zero singular values of $\mathbf{M}$, and $\mathbf{U}_M \in \mathbb{R}^{m \times r}$ and $\mathbf{V}_M \in \mathbb{R}^{n \times r}$ are the corresponding left and right singular vectors of $\mathbf{M}$. We define $\mathbf{M}^+ = \mathbf{V}_M \mathbf{\Sigma}_M^{-1} \mathbf{U}_M^\top$ as the Moore-Penrose pseudoinverse of $\mathbf{M}$ and $\mathbf{P}_M = \mathbf{M}\mathbf{M}^+$ as the orthogonal projection onto the column space of $\mathbf{M}$. We let $\|\cdot\|_2$, $\|\cdot\|_F$, and $\|\cdot\|_*$ respectively denote the spectral, Frobenius, and nuclear norms of a matrix, $\|\cdot\|_\infty$ denote the maximum entry of a matrix, and $\|\cdot\|$ represent the ℓ_2 norm of a vector.

2. The Divide-Factor-Combine Framework

In this section, we present a general divide-and-conquer framework for scalable noisy matrix factorization. We begin by defining the problem setting of interest.

2.1 Noisy Matrix Factorization (MF)

In the setting of noisy matrix factorization, we observe a subset of the entries of a matrix $\mathbf{M} = \mathbf{L}_0 + \mathbf{S}_0 + \mathbf{Z}_0 \in \mathbb{R}^{m \times n}$, where $\mathbf{L}_0$ has rank $r \ll m, n$, $\mathbf{S}_0$ represents a sparse matrix of outliers of arbitrary magnitude, and $\mathbf{Z}_0$ is a dense noise matrix. We let Ω represent the locations of the observed entries and $\mathcal{P}_\Omega$ be the orthogonal projection onto the space of $m \times n$ matrices with support Ω, so that

$$(\mathcal{P}_\Omega(\mathbf{M}))_{ij} = \mathbf{M}_{ij}, \text{ if } (i,j) \in \Omega \quad \text{and} \quad (\mathcal{P}_\Omega(\mathbf{M}))_{ij} = 0 \text{ otherwise.}^2$$

Our goal is to estimate the low-rank matrix $\mathbf{L}_0$ from $\mathcal{P}_\Omega(\mathbf{M})$ with error proportional to the noise level $\Delta \triangleq \|\mathbf{Z}_0\|_F$. We will focus on two specific instances of this general problem:

- **Noisy Matrix Completion (MC):** $s \triangleq |\Omega|$ entries of $\mathbf{M}$ are revealed uniformly without replacement, along with their locations. There are no outliers, so that $\mathbf{S}_0$ is identically zero.
- **Noisy Robust Matrix Factorization (RMF):** $\mathbf{S}_0$ is identically zero save for s outlier entries of arbitrary magnitude with unknown locations distributed uniformly without replacement. All entries of $\mathbf{M}$ are observed, so that $\mathcal{P}_\Omega(\mathbf{M}) = \mathbf{M}$.

2.2 Divide-Factor-Combine

The Divide-Factor-Combine (DFC) framework divides the expensive task of matrix factorization into smaller subproblems, executes those subproblems in parallel, and then efficiently combines the results into a final low-rank estimate of $\mathbf{L}_0$. We highlight three variants of this general framework in Algorithms 1, 2, and 3. These algorithms, which we refer to as DFC-Proj, DFC-RP, and DFC-Nys, differ in their strategies for division and recombination but adhere to a common pattern of three simple steps:

(D step) Divide input matrix into submatrices: DFC-Proj and DFC-RP randomly partition $\mathcal{P}_\Omega(\mathbf{M})$ into t l-column submatrices, $\{\mathcal{P}_\Omega(\mathbf{C}_1), \ldots, \mathcal{P}_\Omega(\mathbf{C}_t)\}$,[3] while DFC-Nys selects an l-column submatrix, $\mathcal{P}_\Omega(\mathbf{C})$, and a d-row submatrix, $\mathcal{P}_\Omega(\mathbf{R})$, uniformly at random.

(F step) Factor each submatrix in parallel using any base MF algorithm: DFC-Proj and DFC-RP perform t parallel submatrix factorizations, while DFC-Nys performs two such parallel factorizations. Standard base MF algorithms output the following low-rank approximations: $\{\hat{\mathbf{C}}_1, \ldots, \hat{\mathbf{C}}_t\}$ for DFC-Proj and DFC-RP; $\hat{\mathbf{C}}$ and $\hat{\mathbf{R}}$ for DFC-Nys. All matrices are retained in factored form.

(C step) Combine submatrix estimates: DFC-Proj generates a final low-rank estimate $\hat{\mathbf{L}}^{proj}$ by projecting $[\hat{\mathbf{C}}_1, \ldots, \hat{\mathbf{C}}_t]$ onto the column space of $\hat{\mathbf{C}}_1$, DFC-RP uses random projection to compute a rank-k estimate $\hat{\mathbf{L}}^{rp}$ of $[\hat{\mathbf{C}}_1 \cdots \hat{\mathbf{C}}_t]$ where k is the median rank of the returned subproblem estimates, and DFC-Nys forms the low-rank

2. When $\mathbf{Q}$ is a submatrix of $\mathbf{M}$ we abuse notation and let $\mathcal{P}_\Omega(\mathbf{Q})$ be the corresponding submatrix of $\mathcal{P}_\Omega(\mathbf{M})$.
3. For ease of discussion, we assume that t evenly divides n so that $l = n/t$. In general, $\mathcal{P}_\Omega(\mathbf{M})$ can always be partitioned into t submatrices, each with either $\lfloor n/t \rfloor$ or $\lceil n/t \rceil$ columns.

Algorithm 1 DFC-Proj

Input: $\mathcal{P}_\Omega(\mathbf{M})$, t
$\{\mathcal{P}_\Omega(\mathbf{C}_i)\}_{1\leq i\leq t} = \textsc{SampCol}(\mathcal{P}_\Omega(\mathbf{M}),\, t)$
do in parallel
$\hat{\mathbf{C}}_1 = \textsc{Base-MF-Alg}(\mathcal{P}_\Omega(\mathbf{C}_1))$
$\vdots$
$\hat{\mathbf{C}}_t = \textsc{Base-MF-Alg}(\mathcal{P}_\Omega(\mathbf{C}_t))$
end do
$\hat{\mathbf{L}}^{proj} = \textsc{ColProjection}(\hat{\mathbf{C}}_1, \ldots, \hat{\mathbf{C}}_t)$

Algorithm 2 DFC-RP

Input: $\mathcal{P}_\Omega(\mathbf{M})$, t
$\{\mathcal{P}_\Omega(\mathbf{C}_i)\}_{1\leq i\leq t} = \textsc{SampCol}(\mathcal{P}_\Omega(\mathbf{M}),\, t)$
do in parallel
$\hat{\mathbf{C}}_1 = \textsc{Base-MF-Alg}(\mathcal{P}_\Omega(\mathbf{C}_1))$
$\vdots$
$\hat{\mathbf{C}}_t = \textsc{Base-MF-Alg}(\mathcal{P}_\Omega(\mathbf{C}_t))$
end do
$k = \text{median}_{i\in\{1\ldots t\}}\big(\text{rank}(\hat{\mathbf{C}}_i)\big)$
$\hat{\mathbf{L}}^{proj} = \textsc{RandProjection}(\hat{\mathbf{C}}_1, \ldots, \hat{\mathbf{C}}_t, k)$

Algorithm 3 DFC-Nys

Input: $\mathcal{P}_\Omega(\mathbf{M})$, l, d
$\mathcal{P}_\Omega(\mathbf{C}), \mathcal{P}_\Omega(\mathbf{R}) = \textsc{SampColRow}(\mathcal{P}_\Omega(\mathbf{M}),\, l,\, d)$
do in parallel
$\hat{\mathbf{C}} = \textsc{Base-MF-Alg}(\mathcal{P}_\Omega(\mathbf{C}))$
$\hat{\mathbf{R}} = \textsc{Base-MF-Alg}(\mathcal{P}_\Omega(\mathbf{R}))$
end do
$\hat{\mathbf{L}}^{nys} = \textsc{GenNyström}(\hat{\mathbf{C}},\, \hat{\mathbf{R}})$

estimate $\hat{\mathbf{L}}^{nys}$ from $\hat{\mathbf{C}}$ and $\hat{\mathbf{R}}$ via the generalized Nyström method. These matrix approximation techniques are described in more detail in Section 2.3.

2.3 Randomized Matrix Approximations

Underlying the C step of each DFC algorithm is a method for generating randomized low-rank approximations to an arbitrary matrix $\mathbf{M}$.

Column Projection: DFC-Proj (Algorithm 1) uses the column projection method of Frieze et al. (1998). Suppose that $\mathbf{C}$ is a matrix of l columns sampled uniformly and without replacement from the columns of $\mathbf{M}$. Then, column projection generates a "matrix projection" approximation (Kumar et al., 2009a) of $\mathbf{M}$ via

$$\mathbf{L}^{proj} = \mathbf{C}\mathbf{C}^{+}\mathbf{M} = \mathbf{U}_C\mathbf{U}_C^{\top}\mathbf{M}.$$

In practice, we do not reconstruct $\mathbf{L}^{proj}$ but rather maintain low-rank factors, e.g., $\mathbf{U}_C$ and $\mathbf{U}_C^{\top}\mathbf{M}$.

Random Projection: The celebrated result of Johnson and Lindenstrauss (1984) shows that random low-dimensional embeddings preserve Euclidean geometry. Inspired by this result, several random projection algorithms (e.g., Papadimitriou et al., 1998; Liberty, 2009; Rokhlin et al., 2009) have been introduced for approximating a matrix by projecting it onto a random low-dimensional subspace (see Halko et al. 2011 for further discussion). DFC-RP (Algorithm 2) uses such a random projection method due to Halko et al. (2011). Given a

target low-rank parameter k, let $\mathbf{G}$ be an $n \times (k+p)$ standard Gaussian matrix $\mathbf{G}$, where p is an oversampling parameter. Next, let $\mathbf{Y} = (\mathbf{M}\mathbf{M}^\top)^q\mathbf{M}\mathbf{G}$, and define $\mathbf{Q} \in \mathbb{R}^{m\times k}$ as the top k left singular vectors of $\mathbf{Y}$. The random projection approximation of $\mathbf{M}$ is then given by

$$\mathbf{L}^{rp} = \mathbf{Q}\mathbf{Q}^+\mathbf{M}.$$

We work with an implementation (Tygert, 2009) of a numerically stable variant of this algorithm described in Algorithm 4.4 of Halko et al. (2011). Moreover, the parameters p and q are typically set to small positive constants (Tygert, 2009; Halko et al., 2011), and we set $p = 5$ and $q = 2$.

Generalized Nyström Method: The Nyström method was developed for the discretization of integral equations (Nyström, 1930) and has since been used to speed up large-scale learning applications involving symmetric positive semidefinite matrices (Williams and Seeger, 2000). DFC-Nys (Algorithm 3) makes use of a generalization of the Nyström method for arbitrary real matrices (Goreinov et al., 1997). Suppose that $\mathbf{C}$ consists of l columns of $\mathbf{M}$, sampled uniformly without replacement, and that $\mathbf{R}$ consists of d rows of $\mathbf{M}$, independently sampled uniformly and without replacement. Let $\mathbf{W}$ be the $d \times l$ matrix formed by sampling the corresponding rows of $\mathbf{C}$.[4] Then, the generalized Nyström method computes a "spectral reconstruction" approximation (Kumar et al., 2009a) of $\mathbf{M}$ via

$$\mathbf{L}^{nys} = \mathbf{C}\mathbf{W}^+\mathbf{R} = \mathbf{C}\mathbf{V}_W\mathbf{\Sigma}_W^+\mathbf{U}_W^\top\mathbf{R}\,.$$

As with $\mathbf{M}^{proj}$, we store low-rank factors of $\mathbf{L}^{nys}$, such as $\mathbf{C}\mathbf{V}_W\mathbf{\Sigma}_W^+$ and $\mathbf{U}_W^\top\mathbf{R}$.

Algorithm	**Factorization (Per Iteration)**		**Combine Step**	
	Serial	**Parallel**	**Serial**	**Parallel**
Base Alg	$O(mn\hat{k})$	$O(mn\hat{k})$	-	-
DFC-Proj	$O(tml\hat{k})$	$O(ml\hat{k})$	$O(tm\hat{k}^2)$	$O(m\hat{k}^2)$
DFC-RP	$O(tml\hat{k})$	$O(ml\hat{k})$	$O(tm\hat{k}^2 + n\hat{k})$	$O(m\hat{k}^2 + tm\hat{k} + n\hat{k})$
DFC-Nys	$O((ml + nd)\hat{k})$	$O(\max(ml, nd)\hat{k})$	$O(m\hat{k}^2)$	$O(m\hat{k}^2)$

Table 1: Summary of running time complexity of DFC variants in contrast to many standard start-of-the-art MF algorithms. This running time analysis assumes that $l \leq m \leq n$ and that all low-rank matrices considered have rank $\hat{k}$. See Section 2.4 for a more detailed analysis.

2.4 Running Time of DFC

Many state-of-the-art MF algorithms have $\Omega(mnk_M)$ per-iteration time complexity due to the rank-k_M truncated SVD performed on each iteration. DFC significantly reduces the per-iteration complexity to $O(mlk_{C_i})$ time for $\mathbf{C}_i$ (or $\mathbf{C}$) and $O(ndk_R)$ time for $\mathbf{R}$. The cost of combining the submatrix estimates is even smaller when using column projection or the generalized Nyström method, since the outputs of standard MF algorithms are returned

4. This choice is arbitrary: $\mathbf{W}$ could also be defined as a submatrix of $\mathbf{R}$.

in factored form. Indeed, if we define $k' \triangleq \max_i k_{C_i}$, then the column projection step of DFC-Proj requires only $O(mk'^2 + lk'^2)$ time: $O(mk'^2 + lk'^2)$ time for the pseudoinversion of $\hat{\mathbf{C}}_1$ and $O(mk'^2 + lk'^2)$ time for matrix multiplication with each $\hat{\mathbf{C}}_i$ in parallel. Similarly, the generalized Nyström step of DFC-Nys requires only $O(l\bar{k}^2 + d\bar{k}^2 + \min(m,n)\bar{k}^2)$ time, where $\bar{k} \triangleq \max(k_C, k_R)$.

DFC-RP also benefits from the factored form of the outputs of standard MF algorithms. Assuming that p and q are positive constants, the random projection step of DFC-RP requires $O(mkt + mkk' + lkk' + nk)$ time where k is the low-rank parameter of $\mathbf{Q}$: $O(nk)$ time to generate $\mathbf{G}$, $O(mkk' + lkk' + mkt)$ to compute $\mathbf{Y}$ in parallel, $O(mk^2)$ to compute the SVD of $\mathbf{Y}$, and $O(mk'^2 + lk'^2)$ time for matrix multiplication with each $\hat{\mathbf{C}}_i$ in parallel in the final projection step. Note that the running time of the random projection step depends on t (even when executed in parallel) and thus has a larger complexity than the column projection and generalized Nyström variants. Nevertheless, the random projection step need be performed only once and thus yields a significant savings over the repeated computation of SVDs required by typical base algorithms.

A summary of these running times is presented in Table 1.

2.5 Ensemble Methods

Ensemble methods have been shown to improve performance of matrix approximation algorithms, while straightforwardly leveraging the parallelism of modern many-core and distributed architectures (Kumar et al., 2009b). As such, we propose ensemble variants of the DFC algorithms that demonstrably reduce estimation error while introducing a negligible cost to the parallel running time. For DFC-Proj-Ens, rather than projecting only onto the column space of $\hat{\mathbf{C}}_1$, we project $[\hat{\mathbf{C}}_1, \ldots, \hat{\mathbf{C}}_t]$ onto the column space of each $\hat{\mathbf{C}}_i$ in parallel and then average the t resulting low-rank approximations. For DFC-RP-Ens, rather than projecting only onto a column space derived from a single random matrix $\mathbf{G}$, we project $[\hat{\mathbf{C}}_1, \ldots, \hat{\mathbf{C}}_t]$ onto t column spaces derived from t random matrices in parallel and then average the t resulting low-rank approximations. For DFC-Nys-Ens, we choose a random d-row submatrix $\mathcal{P}_\Omega(\mathbf{R})$ as in DFC-Nys and independently partition the columns of $\mathcal{P}_\Omega(\mathbf{M})$ into $\{\mathcal{P}_\Omega(\mathbf{C}_1), \ldots, \mathcal{P}_\Omega(\mathbf{C}_t)\}$ as in DFC-Proj and DFC-RP. After running the base MF algorithm on each submatrix, we apply the generalized Nyström method to each $(\hat{\mathbf{C}}_i, \hat{\mathbf{R}})$ pair in parallel and average the t resulting low-rank approximations. Section 6 highlights the empirical effectiveness of ensembling.

3. Roadmap of Theoretical Analysis

While DFC in principle can work with any base matrix factorization algorithm, it offers the greatest benefits when united with accurate but computationally expensive base procedures. Convex optimization approaches to matrix completion and robust matrix factorization (e.g., Lin et al., 2009b; Ma et al., 2011; Toh and Yun, 2010) are prime examples of this class, since they admit strong theoretical estimation guarantees (Agarwal et al., 2011; Candès et al., 2011; Candès and Plan, 2010; Negahban and Wainwright, 2012) but suffer from poor computational complexity due to the repeated and costly computation of truncated SVDs. Section 6 will provide empirical evidence that DFC provides an attractive framework to

improve the scalability of these algorithms, but we first present a thorough theoretical analysis of the estimation properties of DFC.

Over the course of the next three sections, we will show that the same assumptions that give rise to strong estimation guarantees for standard MF formulations also guarantee strong estimation properties for DFC. While these results represent an important first step toward understanding the theoretical behavior of DFC, we will see that certain gaps remain between our theoretical characterization and the practical performance of DFC. We will reflect on these gaps and the attendant opportunities for tightened theoretical analysis in Section 6.4. In the remainder of this section, we first introduce these standard assumptions and then present simplified bounds to build intuition for our theoretical results and our underlying proof techniques.

3.1 Standard Assumptions for Noisy Matrix Factorization

Since not all matrices can be recovered from missing entries or gross outliers, recent theoretical advances have studied sufficient conditions for accurate noisy MC (Candès and Plan, 2010; Keshavan et al., 2010; Negahban and Wainwright, 2012) and RMF (Agarwal et al., 2011; Zhou et al., 2010). Informally, these conditions capture the degree to which information about a single entry is "spread out" across a matrix. The ease of matrix estimation is correlated with this spread of information. The most prevalent set of conditions are *matrix coherence* conditions, which limit the extent to which the singular vectors of a matrix are correlated with the standard basis. However, there exist classes of matrices that violate the coherence conditions but can nonetheless be recovered from missing entries or gross outliers. Negahban and Wainwright (2012) define an alternative notion of *matrix spikiness* in part to handle these classes.

3.1.1 Matrix Coherence

Letting $\mathbf{e}_i$ be the ith column of the standard basis, we define two standard notions of coherence (Recht, 2011):

Definition 1 (μ_0-Coherence) *Let $\mathbf{V} \in \mathbb{R}^{n \times r}$ contain orthonormal columns with $r \leq n$. Then the μ_0-coherence of $\mathbf{V}$ is:*

$$\mu_0(\mathbf{V}) \triangleq \frac{n}{r} \max_{1 \leq i \leq n} \|\mathbf{P}_V \mathbf{e}_i\|^2 = \frac{n}{r} \max_{1 \leq i \leq n} \|\mathbf{V}_{(i)}\|^2 .$$

Definition 2 (μ_1-Coherence) *Let $\mathbf{L} \in \mathbb{R}^{m \times n}$ have rank r. Then, the μ_1-coherence of $\mathbf{L}$ is:*

$$\mu_1(\mathbf{L}) \triangleq \sqrt{\tfrac{mn}{r}} \max_{ij} |\mathbf{e}_i^\top \mathbf{U}_L \mathbf{V}_L^\top \mathbf{e}_j| .$$

For conciseness, we extend the definition of μ_0-coherence to an arbitrary matrix $\mathbf{L} \in \mathbb{R}^{m \times n}$ with rank r via $\mu_0(\mathbf{L}) \triangleq \max(\mu_0(\mathbf{U}_L), \mu_0(\mathbf{V}_L))$. Further, for any $\mu > 0$, we will call a matrix $\mathbf{L}$ (μ, r)-*coherent* if $\text{rank}(\mathbf{L}) = r$, $\mu_0(\mathbf{L}) \leq \mu$, and $\mu_1(\mathbf{L}) \leq \sqrt{\mu}$. Our analysis in Section 4 will focus on base MC and RMF algorithms that express their estimation guarantees in terms of the (μ, r)-coherence of the target low-rank matrix $\mathbf{L}_0$. For such algorithms, lower values of μ correspond to better estimation properties.

3.1.2 Matrix Spikiness

The matrix spikiness condition of Negahban and Wainwright (2012) captures the intuition that a matrix is easier to estimate if its maximum entry is not much larger than its average entry (in the root mean square sense):

Definition 3 (Spikiness) *The spikiness of* $\mathbf{L} \in \mathbb{R}^{m \times n}$ *is:*

$$\alpha(\mathbf{L}) \triangleq \sqrt{mn}\|\mathbf{L}\|_\infty / \|\mathbf{L}\|_F.$$

We call a matrix α*-spiky if* $\alpha(\mathbf{L}) \leq \alpha$.

Our analysis in Section 5 will focus on base MC algorithms that express their estimation guarantees in terms of the α-spikiness of the target low-rank matrix $\mathbf{L}_0$. For such algorithms, lower values of α correspond to better estimation properties.

3.2 Prototypical Estimation Bounds

We now present a prototypical estimation bound for DFC. Suppose that a base MC algorithm solves the *noisy nuclear norm heuristic*, studied in Candès and Plan (2010):

$$\text{minimize}_{\mathbf{L}} \quad \|\mathbf{L}\|_* \quad \text{subject to} \quad \|\mathcal{P}_\Omega(\mathbf{M} - \mathbf{L})\|_F \leq \Delta,$$

and that, for simplicity, $\mathbf{M}$ is square. The following prototype bound, derived from a new noisy MC guarantee in Theorem 10, describes the behavior of this estimator under matrix coherence assumptions. Note that the bound implies exact recovery in the noiseless setting, i.e., when $\Delta = 0$.

Proto-Bound 1 (MC under Incoherence) *Suppose that* $\mathbf{L}_0$ *is* (μ, r)*-coherent,* s *entries of* $\mathbf{M} \in \mathbb{R}^{n \times n}$ *are observed uniformly at random where* $s = \Omega(\mu r n \log^2(n))$*, and* $\|\mathbf{M} - \mathbf{L}_0\|_F \leq \Delta$*. If* $\hat{\mathbf{L}}$ *solves the noisy nuclear norm heuristic, then*

$$\|\mathbf{L}_0 - \hat{\mathbf{L}}\|_F \leq f(n)\Delta$$

with high probability, where f *is a function of* n.

Now we present a corresponding prototype bound for DFC-Proj, a simplified version of our Corollary 14, under precisely the same coherence assumptions. Notably, this bound i) preserves accuracy with a flexible $(2 + \epsilon)$ degradation in estimation error over the base algorithm, ii) allows for speed-up by requiring only a vanishingly small fraction of columns to be sampled (i.e., $l/n \to 0$) whenever $s = \omega(n \log^2(n))$ entries are revealed, and iii) maintains exact recovery in the noiseless setting.

Proto-Bound 2 (DFC-MC under Incoherence) *Suppose that* $\mathbf{L}_0$ *is* (μ, r)*-coherent,* s *entries of* $\mathbf{M} \in \mathbb{R}^{n \times n}$ *are observed uniformly at random, and* $\|\mathbf{M} - \mathbf{L}_0\|_F \leq \Delta$*. Then it suffices to choose*

$$l \geq c \frac{\mu^2 r^2 n^2 \log^2(n)}{s\epsilon^2}$$

random columns suffice to have

$$\|\mathbf{L}_0 - \hat{\mathbf{L}}^{proj}\|_F \leq (2+\epsilon) f(n) \Delta$$

with high probability when the noisy nuclear norm heuristic is used as a base algorithm, where f is the same function of n defined in Proto. 1 and c is a fixed positive constant.

The proof of Proto. 2, and indeed of each of our main DFC results, consists of three high-level steps:

1. *Bound coherence of submatrices*: Recall that the F step of DFC operates by applying a base MF algorithm to submatrices. We show that, with high probability, uniformly sampled submatrices are only moderately more coherent and moderately more spiky than the matrix from which they are drawn. This allows for accurate estimation of submatrices using base algorithms with standard coherence or spikiness requirements. The conservation of incoherence result is summarized in Lemma 4, while the conservation of non-spikiness is presented in Lemma 17.

2. *Bound error of randomized matrix approximations*: The error introduced by the C step of DFC depends on the framework variant. Drawing upon tools from randomized ℓ_2 regression (Drineas et al., 2008), randomized matrix multiplication (Drineas et al., 2006a,b), and matrix concentration (Hsu et al., 2012), we show that the same assumptions on the spread of information responsible for accurate MC and RMF also yield high fidelity reconstructions for column projection (Corollary 6 and Theorem 18) and the Nyström method (Corollary 7 and Corollary 8). We additionally present general approximation guarantees for random projection due to Halko et al. (2011) in Corollary 9. These results give rise to "master theorems" for coherence (Theorem 12) and spikiness (Theorem 20) that generically relate the estimation error of DFC to the error of any base algorithm.

3. *Bound error of submatrix factorizations*: The final step combines a master theorem with a base estimation guarantee applied to each DFC subproblem. We study both new (Theorem 10) and established bounds (Theorem 11 and Corollary 19) for MC and RMF and prove that DFC submatrices satisfy the base guarantee preconditions with high probability. We present the resulting coherence-based estimation guarantees for DFC in Corollary 14 and Corollary 16 and the spikiness-based estimation guarantee in Corollary 22.

The next two sections present the main results contributing to each of these proof steps, as well as their consequences for MC and RMF. Section 4 presents our analysis under coherence assumptions, while Section 5 contains our spikiness analysis.

4. Coherence-based Theoretical Analysis

This section presents our analysis of DFC under standard coherence assumptions encountered in the MC and RMF literature.

4.1 Coherence Analysis of Randomized Approximation Algorithms

We begin our coherence-based analysis by characterizing the behavior of randomized approximation algorithms under standard coherence assumptions. The derived properties will aid us in deriving DFC estimation guarantees. Hereafter, $\epsilon \in (0, 1]$ represents a prescribed error tolerance, and $\delta, \delta' \in (0, 1]$ denote target failure probabilities.

4.1.1 Conservation of Incoherence

Our first result bounds the μ_0 and μ_1-coherence of a uniformly sampled submatrix in terms of the coherence of the full matrix. This conservation of incoherence allows for accurate submatrix completion or submatrix outlier removal when using standard MC and RMF algorithms. Its proof is given in Section B.

Lemma 4 (Conservation of Incoherence) *Let $\mathbf{L} \in \mathbb{R}^{m\times n}$ be a rank-r matrix and define $\mathbf{L}_C \in \mathbb{R}^{m\times l}$ as a matrix of l columns of $\mathbf{L}$ sampled uniformly without replacement. If $l \geq cr\mu_0(\mathbf{V}_L)\log(n)\log(1/\delta)/\epsilon^2$, where c is a fixed positive constant defined in Corollary 6, then*

i) $\operatorname{rank}(\mathbf{L}_C) = \operatorname{rank}(\mathbf{L})$

ii) $\mu_0(\mathbf{U}_{L_C}) = \mu_0(\mathbf{U}_L)$

iii) $\mu_0(\mathbf{V}_{L_C}) \leq \dfrac{\mu_0(\mathbf{V}_L)}{1-\epsilon/2}$

iv) $\mu_1^2(\mathbf{L}_C) \leq \dfrac{r\mu_0(\mathbf{U}_L)\mu_0(\mathbf{V}_L)}{1-\epsilon/2}$

all hold jointly with probability at least $1 - \delta/n$.

4.1.2 Column Projection Analysis

Our next result shows that projection based on uniform column sampling leads to near optimal estimation in matrix regression when the covariate matrix has small coherence. This statement will immediately give rise to estimation guarantees for column projection and the generalized Nyström method.

Theorem 5 (Subsampled Regression under Incoherence) *Given a target matrix $\mathbf{B} \in \mathbb{R}^{p\times n}$ and a rank-r matrix of covariates $\mathbf{L} \in \mathbb{R}^{m\times n}$, choose $l \geq 3200r\mu_0(\mathbf{V}_L)\log(4n/\delta)/\epsilon^2$, let $\mathbf{B}_C \in \mathbb{R}^{p\times l}$ be a matrix of l columns of $\mathbf{B}$ sampled uniformly without replacement, and let $\mathbf{L}_C \in \mathbb{R}^{m\times l}$ consist of the corresponding columns of $\mathbf{L}$. Then,*

$$\|\mathbf{B} - \mathbf{B}_C\mathbf{L}_C^{+}\mathbf{L}\|_F \leq (1+\epsilon)\|\mathbf{B} - \mathbf{B}\mathbf{L}^{+}\mathbf{L}\|_F$$

with probability at least $1 - \delta - 0.2$.

Fundamentally, Theorem 5 links the notion of coherence, common in matrix estimation communities, to the randomized approximation concept of *leverage score sampling* (Mahoney and Drineas, 2009). The proof of Theorem 5, given in Section A, builds upon the

randomized ℓ_2 regression work of Drineas et al. (2008) and the matrix concentration results of Hsu et al. (2012) to yield a subsampled regression guarantee with better sampling complexity than that of Drineas et al. (2008, Theorem 5).

A first consequence of Theorem 5 shows that, with high probability, column projection produces an estimate nearly as good as a given rank-r target by sampling a number of columns proportional to the coherence and $r \log n$.

Corollary 6 (Column Projection under Incoherence) *Given a matrix $\mathbf{M} \in \mathbb{R}^{m \times n}$ and a rank-r approximation $\mathbf{L} \in \mathbb{R}^{m \times n}$, choose $l \geq cr\mu_0(\mathbf{V}_L) \log(n) \log(1/\delta)/\epsilon^2$, where c is a fixed positive constant, and let $\mathbf{C} \in \mathbb{R}^{m \times l}$ be a matrix of l columns of $\mathbf{M}$ sampled uniformly without replacement. Then,*

$$\|\mathbf{M} - \mathbf{C}\mathbf{C}^+\mathbf{M}\|_F \leq (1+\epsilon)\|\mathbf{M} - \mathbf{L}\|_F$$

with probability at least $1 - \delta$.

Our result generalizes Theorem 1 of Drineas et al. (2008) by providing improved sampling complexity and guarantees relative to an *arbitrary* low-rank approximation. Notably, in the "noiseless" setting, when $\mathbf{M} = \mathbf{L}$, Corollary 6 guarantees exact recovery of $\mathbf{M}$ with high probability. The proof of Corollary 6 is given in Section C.

4.1.3 Generalized Nyström Analysis

Theorem 5 and Corollary 6 together imply an estimation guarantee for the generalized Nyström method relative to an arbitrary low-rank approximation $\mathbf{L}$. Indeed, if the matrix of sampled columns is denoted by $\mathbf{C}$, then, with appropriately reduced probability, $\mathrm{O}(\mu_0(\mathbf{V}_L) r \log n)$ columns and $\mathrm{O}(\mu_0(\mathbf{U}_C) r \log m)$ rows suffice to match the reconstruction error of $\mathbf{L}$ up to any fixed precision. The proof can be found in Section D.

Corollary 7 (Generalized Nyström under Incoherence) *Given a matrix $\mathbf{M} \in \mathbb{R}^{m \times n}$ and a rank-r approximation $\mathbf{L} \in \mathbb{R}^{m \times n}$, choose $l \geq cr\mu_0(\mathbf{V}_L) \log(n) \log(1/\delta)/\epsilon^2$ with c a constant as in Corollary 6, and let $\mathbf{C} \in \mathbb{R}^{m \times l}$ be a matrix of l columns of $\mathbf{M}$ sampled uniformly without replacement. Further choose $d \geq cl\mu_0(\mathbf{U}_C) \log(m) \log(1/\delta')/\epsilon^2$, and let $\mathbf{R} \in \mathbb{R}^{d \times n}$ be a matrix of d rows of $\mathbf{M}$ sampled independently and uniformly without replacement. Then,*

$$\|\mathbf{M} - \mathbf{C}\mathbf{W}^+\mathbf{R}\|_F \leq (1+\epsilon)^2\|\mathbf{M} - \mathbf{L}\|_F$$

with probability at least $(1 - \delta)(1 - \delta' - 0.2)$.

Like the generalized Nyström bound of Drineas et al. (2008, Theorem 4) and unlike our column projection result, Corollary 7 depends on the coherence of the submatrix $\mathbf{C}$ and holds only with probability bounded away from 1. Our next contribution shows that we can do away with these restrictions in the noiseless setting, where $\mathbf{M} = \mathbf{L}$.

Corollary 8 (Noiseless Generalized Nyström under Incoherence) *Let $\mathbf{L} \in \mathbb{R}^{m \times n}$ be a rank-r matrix. Choose $l \geq 48r\mu_0(\mathbf{V}_L) \log(4n/(1-\sqrt{1-\delta}))$ and $d \geq 48r\mu_0(\mathbf{U}_L) \log(4m/(1-\sqrt{1-\delta}))$. Let $\mathbf{C} \in \mathbb{R}^{m \times l}$ be a matrix of l columns of $\mathbf{L}$ sampled uniformly without replacement, and let $\mathbf{R} \in \mathbb{R}^{d \times n}$ be a matrix of d rows of $\mathbf{L}$ sampled independently and uniformly without replacement. Then,*

$$\mathbf{L} = \mathbf{C}\mathbf{W}^+\mathbf{R}$$

with probability at least $1 - \delta$.

This result may appear surprising at first sight, since only vanishingly small fractions of rows and columns may participate in the generalized Nyström reconstruction. The intuition for the method's success that when the rank of $\mathbf{L}$ is small, only a small number of well-chosen rows and columns are needed to reconstruct the row and column space of $\mathbf{L}$ and that, when $\mathbf{L}$ is incoherent, uniform random sampling is likely produce well-chosen rows and columns. The proof of Corollary 8, given in Section E, adapts a strategy of Talwalkar and Rostamizadeh (2010) developed for the analysis of positive semidefinite matrices.

4.1.4 Random Projection Analysis

We next present an estimation guarantee for the random projection method relative to an arbitrary low-rank approximation $\mathbf{L}$. The result implies that using a random matrix with oversampled columns proportional to $r \log(1/\delta)$ suffices to match the reconstruction error of $\mathbf{L}$ up to any fixed precision with probability $1 - \delta$. The result is a direct consequence of the random projection analysis of Halko et al. (2011, Theorem 10.7), and the proof can be found in Section F.

Corollary 9 (Random Projection) *Given a matrix* $\mathbf{M} \in \mathbb{R}^{m \times n}$ *and a rank-r approximation* $\mathbf{L} \in \mathbb{R}^{m \times n}$ *with* $r \geq 2$*, choose an oversampling parameter*

$$p \geq 242 \; r \log(7/\delta)/\epsilon^2.$$

Draw an $n \times (r + p)$ *standard Gaussian matrix* $\mathbf{G}$ *and define* $\mathbf{Y} = \mathbf{MG}$*. Then, with probability at least* $1 - \delta$*,*

$$\|\mathbf{M} - \mathbf{P}_Y \mathbf{M}\|_F \leq (1 + \epsilon)\|\mathbf{M} - \mathbf{L}\|_F.$$

Moreover, define $\mathbf{L}^{rp}$ *as the best rank-r approximation of* $\mathbf{P}_Y \mathbf{M}$ *with respect to the Frobenius norm. Then, with probability at least* $1 - \delta$*,*

$$\|\mathbf{M} - \mathbf{L}^{rp}\|_F \leq (2 + \epsilon)\|\mathbf{M} - \mathbf{L}\|_F.$$

We note that, in contrast to Corollary 6 and Corollary 7, Corollary 9 does not depend on the coherence of $\mathbf{L}$ and hence can be fruitfully applied even in the absence of an incoherence assumption. We demonstrate such a use case in Section 5. We note moreover that past empirical studies have demonstrated excellent estimation error with $p \leq 10$ irrespective of the target matrix rank (Halko et al., 2011); bridging the gap between theory and practice in this instance represents an interesting open problem.

4.2 Base Algorithm Guarantees

As prototypical examples of the coherence-based estimation guarantees available for noisy MC and noisy RMF, consider the following two theorems. The first bounds the estimation error of a convex optimization approach to noisy matrix completion, under the assumptions of incoherence and uniform sampling.

Theorem 10 (Noisy MC under Incoherence) *Suppose that* $\mathbf{L}_0 \in \mathbb{R}^{m \times n}$ *is* (μ, r)*-coherent and that, for some target rate parameter* $\beta > 1$*,*

$$s \geq 32\mu r(m+n)\beta \log^2(m+n)$$

entries of $\mathbf{M}$ *are observed with locations* Ω *sampled uniformly without replacement. Then, if* $m \leq n$ *and* $\|\mathcal{P}_\Omega(\mathbf{M}) - \mathcal{P}_\Omega(\mathbf{L}_0)\|_F \leq \Delta$ *a.s., the minimizer* $\hat{\mathbf{L}}$ *of the problem*

$$\text{minimize}_{\mathbf{L}} \quad \|\mathbf{L}\|_* \quad \text{subject to} \quad \|\mathcal{P}_\Omega(\mathbf{M} - \mathbf{L})\|_F \leq \Delta. \tag{1}$$

satisfies

$$\|\mathbf{L}_0 - \hat{\mathbf{L}}\|_F \leq 8\sqrt{\frac{2m^2n}{s} + m + \frac{1}{16}}\Delta \leq c_e\sqrt{mn}\Delta$$

with probability at least $1 - 4\log(n)n^{2-2\beta}$ *for* c_e *a positive constant.*

A similar estimation guarantee was obtained by Candès and Plan (2010) under stronger assumptions. We give the proof of Theorem 10 in Section J.

The second result, due to Zhou et al. (2010) and reformulated for a generic rate parameter β, as described in Candès et al. (2011, Section 3.1), bounds the estimation error of a convex optimization approach to noisy RMF, under the assumptions of incoherence and uniformly distributed outliers.

Theorem 11 (Noisy RMF under Incoherence, Zhou et al. 2010, Theorem 2) *Suppose that* $\mathbf{L}_0$ *is* (μ, r)*-coherent and that the support set of* $\mathbf{S}_0$ *is uniformly distributed among all sets of cardinality* s*. Then, if* $m \leq n$ *and* $\|\mathbf{M} - \mathbf{L}_0 - \mathbf{S}_0\|_F \leq \Delta$ *a.s., there is a constant* c_p *such that with probability at least* $1 - c_p n^{-\beta}$*, the minimizer* $(\hat{\mathbf{L}}, \hat{\mathbf{S}})$ *of the problem*

$$\text{minimize}_{\mathbf{L},\mathbf{S}} \quad \|\mathbf{L}\|_* + \lambda\|\mathbf{S}\|_1 \quad \text{subject to} \quad \|\mathbf{M} - \mathbf{L} - \mathbf{S}\|_F \leq \Delta \quad \text{with} \quad \lambda = 1/\sqrt{n} \tag{2}$$

satisfies $\|\mathbf{L}_0 - \hat{\mathbf{L}}\|_F^2 + \|\mathbf{S}_0 - \hat{\mathbf{S}}\|_F^2 \leq c_e'^2 mn\Delta^2$*, provided that*

$$r \leq \frac{\rho_r m}{\mu \log^2(n)} \quad \text{and} \quad s \leq (1 - \rho_s\beta)mn$$

for target rate parameter $\beta > 2$*, and positive constants* ρ_r, ρ_s*, and* c_e'*.*

4.3 Coherence Master Theorem

We now show that the same coherence conditions that allow for accurate MC and RMF also imply high-probability estimation guarantees for DFC. To make this precise, we let $\mathbf{M} = \mathbf{L}_0 + \mathbf{S}_0 + \mathbf{Z}_0 \in \mathbb{R}^{m \times n}$, where $\mathbf{L}_0$ is (μ, r)-coherent and $\|\mathcal{P}_\Omega(\mathbf{Z}_0)\|_F \leq \Delta$. Then, our next theorem provides a generic bound on the estimation error of DFC used in combination with an arbitrary base algorithm. The proof, which builds upon the results of Section 4.1, is given in Section G.

Theorem 12 (Coherence Master Theorem) *Choose $t = n/l$, $l \geq cr\mu \log(n)\log(2/\delta)/\epsilon^2$, where c is a fixed positive constant, and $p \geq 242\ r\log(14/\delta)/\epsilon^2$. Under the notation of Algorithms 1 and 2, let $\{\mathbf{C}_{0,1}, \cdots, \mathbf{C}_{0,t}\}$ be the corresponding partition of $\mathbf{L}_0$. Then, with probability at least $1-\delta$, $\mathbf{C}_{0,i}$ is $(\frac{r\mu^2}{1-\epsilon/2}, r)$-coherent for all i, and*

$$\|\mathbf{L}_0 - \hat{\mathbf{L}}^*\|_F \leq (2+\epsilon)\sqrt{\textstyle\sum_{i=1}^t \|\mathbf{C}_{0,i} - \hat{\mathbf{C}}_i\|_F^2},$$

where $\hat{\mathbf{L}}^*$ is the estimate returned by either DFC-Proj *or* DFC-RP*.*

Under the notation of Algorithm 3, let $\mathbf{C}_0$ and $\mathbf{R}_0$ be the corresponding column and row submatrices of $\mathbf{L}_0$. If in addition $d \geq cl\mu_0(\hat{\mathbf{C}})\log(m)\log(4/\delta)/\epsilon^2$, then, with probability at least $(1-\delta)(1-\delta-0.2)$, DFC-Nys *guarantees that $\mathbf{C}_0$ and $\mathbf{R}_0$ are $(\frac{r\mu^2}{1-\epsilon/2}, r)$-coherent and that*

$$\|\mathbf{L}_0 - \hat{\mathbf{L}}^{nys}\|_F \leq (2+3\epsilon)\sqrt{\|\mathbf{C}_0 - \hat{\mathbf{C}}\|_F^2 + \|\mathbf{R}_0 - \hat{\mathbf{R}}\|_F^2}.$$

Remark 13 *The* DFC-Nys *guarantee requires the number of rows sampled to grow in proportion to $\mu_0(\hat{\mathbf{C}})$, a quantity always bounded by μ in our simulations. Here and in the consequences to follow, the* DFC-Nys *result can be strengthened in the noiseless setting ($\Delta = 0$) by utilizing Corollary 8 in place of Corollary 7 in the proof of Theorem 12.*

When a target matrix is incoherent, Theorem 12 asserts that – with high probability for DFC-Proj and DFC-RP and with fixed probability for DFC-Nys – the estimation error of DFC is not much larger than the error sustained by the base algorithm on each subproblem. Because Theorem 12 further bounds the coherence of each submatrix, we can use any coherence-based matrix estimation guarantee to control the estimation error on each subproblem. The next two sections demonstrate how Theorem 12 can be applied to derive specific DFC estimation guarantees for noisy MC and noisy RMF. In these sections, we let $\bar{n} \triangleq \max(m,n)$.

4.4 Consequences for Noisy MC

As a first consequence of Theorem 12, we will show that DFC retains the high-probability estimation guarantees of a standard MC solver while operating on matrices of much smaller dimension. Suppose that a base MC algorithm solves the convex optimization problem of Eq. (1). Then, Corollary 14 follows from the Coherence Master Theorem (Theorem 12) and the base algorithm guarantee of Theorem 10.

Corollary 14 (DFC-MC under Incoherence) *Suppose that $\mathbf{L}_0$ is (μ, r)-coherent and that s entries of $\mathbf{M}$ are observed, with locations Ω distributed uniformly. Fix any target rate parameter $\beta > 1$. Then, if $\|\mathcal{P}_\Omega(\mathbf{M}) - \mathcal{P}_\Omega(\mathbf{L}_0)\|_F \leq \Delta$ a.s., and the base algorithm solves the optimization problem of Eq. (1), it suffices to choose $t = n/l$,*

$$l \geq c\mu^2 r^2 (m+n) n\beta \log^2(m+n)/(s\epsilon^2), \quad d \geq cl\mu_0(\hat{\mathbf{C}})(2\beta-1)\log^2(4\bar{n})\bar{n}/(n\epsilon^2),$$

and $p \geq 242\ r\log(14\bar{n}^{2\beta-2})/\epsilon^2$ to achieve

$$\textbf{DFC-Proj}: \ \|\mathbf{L}_0 - \hat{\mathbf{L}}^{proj}\|_F \leq (2+\epsilon)c_e\sqrt{mn}\Delta$$

DFC-RP : $\|\mathbf{L}_0 - \hat{\mathbf{L}}^{rp}\|_F \leq (2+\epsilon)c_e\sqrt{mn}\Delta$

DFC-Nys : $\|\mathbf{L}_0 - \hat{\mathbf{L}}^{nys}\|_F \leq (2+3\epsilon)c_e\sqrt{ml+dn}\Delta$

with probability at least

DFC-Proj / DFC-RP : $1 - (5t\log(\bar{n})+1)\bar{n}^{2-2\beta} \geq 1 - \bar{n}^{3-2\beta}$

DFC-Nys : $1 - (10\log(\bar{n})+2)\bar{n}^{2-2\beta} - 0.2$,

respectively, with c as in Theorem 12 and c_e as in Theorem 10.

Remark 15 *Corollary 14 allows for the fraction of columns and rows sampled to decrease as the number of revealed entries, s, increases. Only a vanishingly small fraction of columns ($l/n \to 0$) and rows ($d/\bar{n} \to 0$) need be sampled whenever $s = \omega((m+n)\log^2(m+n))$.*

To understand the conclusions of Corollary 14, consider the base algorithm of Theorem 10, which, when applied to $\mathcal{P}_\Omega(\mathbf{M})$, recovers an estimate $\hat{\mathbf{L}}$ satisfying $\|\mathbf{L}_0 - \hat{\mathbf{L}}\|_F \leq c_e\sqrt{mn}\Delta$ with high probability. Corollary 14 asserts that, with appropriately reduced probability, DFC-Proj and DFC-RP exhibit the same estimation error scaled by an adjustable factor of $2+\epsilon$, while DFC-Nys exhibits a somewhat smaller error scaled by $2+3\epsilon$.

The key take-away is that DFC introduces a controlled increase in error and a controlled decrement in the probability of success, allowing the user to interpolate between maximum speed and maximum accuracy. Thus, DFC can quickly provide near-optimal estimation in the noisy setting and exact recovery in the noiseless setting ($\Delta = 0$), even when entries are missing. The proof of Corollary 14 can be found in Section H.

4.5 Consequences for Noisy RMF

Our next corollary shows that DFC retains the high-probability estimation guarantees of a standard RMF solver while operating on matrices of much smaller dimension. Suppose that a base RMF algorithm solves the convex optimization problem of Eq. (2). Then, Corollary 16 follows from the Coherence Master Theorem (Theorem 12) and the base algorithm guarantee of Theorem 11.

Corollary 16 (DFC-RMF under Incoherence) *Suppose that $\mathbf{L}_0$ is (μ, r)-coherent with*

$$r^2 \leq \frac{\min(m,n)\rho_r}{2\mu^2\log^2(\bar{n})}$$

for a positive constant ρ_r. Suppose moreover that the uniformly distributed support set of $\mathbf{S}_0$ has cardinality s. For a fixed positive constant ρ_s, define the undersampling parameter

$$\beta_s \triangleq \left(1 - \frac{s}{mn}\right)/\rho_s,$$

and fix any target rate parameter $\beta > 2$ with rescaling $\beta' \triangleq \beta\log(\bar{n})/\log(m)$ satisfying $4\beta_s - 3/\rho_s \leq \beta' \leq \beta_s$. Then, if $\|\mathbf{M} - \mathbf{L}_0 - \mathbf{S}_0\|_F \leq \Delta$ a.s., and the base algorithm solves

the optimization problem of Eq. (2), it suffices to choose $t = n/l$,

$$l \geq \max\left(\frac{cr^2\mu^2\beta\log^2(2\bar{n})}{\epsilon^2\rho_r}, \frac{4\log(\bar{n})\beta(1-\rho_s\beta_s)}{m(\rho_s\beta_s-\rho_s\beta')^2}\right),$$
$$d \geq \max\left(\frac{cl\mu_0(\hat{\mathbf{C}})\beta\log^2(4\bar{n})}{\epsilon^2}, \frac{4\log(\bar{n})\beta(1-\rho_s\beta_s)}{n(\rho_s\beta_s-\rho_s\beta')^2}\right)$$

and $p \geq 242\ r\log(14\bar{n}^{\beta})/\epsilon^2$ *to have*

$$\textbf{DFC-Proj}: \ \|\mathbf{L}_0 - \hat{\mathbf{L}}^{proj}\|_F \leq (2+\epsilon)c'_e\sqrt{mn}\Delta$$

$$\textbf{DFC-RP}: \ \|\mathbf{L}_0 - \hat{\mathbf{L}}^{rp}\|_F \leq (2+\epsilon)c'_e\sqrt{mn}\Delta$$

$$\textbf{DFC-Nys}: \ \|\mathbf{L}_0 - \hat{\mathbf{L}}^{nys}\|_F \leq (2+3\epsilon)c'_e\sqrt{ml+dn}\Delta$$

with probability at least

$$\textbf{DFC-Proj / DFC-RP}: \ 1-(t(c_p+1)+1)\bar{n}^{-\beta} \geq 1-c_p\bar{n}^{1-\beta}$$

$$\textbf{DFC-Nys}: \ 1-(2c_p+3)\bar{n}^{-\beta}-0.2,$$

respectively, with c *as in Theorem 12 and* ρ_r, c'_e, *and* c_p *as in Theorem 11.*

Note that Corollary 16 places only very mild restrictions on the number of columns and rows to be sampled. Indeed, l and d need only grow poly-logarithmically in the matrix dimensions to achieve estimation guarantees comparable to those of the RMF base algorithm (Theorem 11). Hence, DFC can quickly provide near-optimal estimation in the noisy setting and exact recovery in the noiseless setting ($\Delta = 0$), even when entries are grossly corrupted. The proof of Corollary 16 can be found in Section I.

5. Theoretical Analysis under Spikiness Conditions

This section presents our analysis of DFC under standard spikiness assumptions from the MC and RMF literature.

5.1 Spikiness Analysis of Randomized Approximation Algorithms

We begin our spikiness analysis by characterizing the behavior of randomized approximation algorithms under standard spikiness assumptions. The derived properties will aid us in developing DFC estimation guarantees. Hereafter, $\epsilon \in (0,1]$ represents a prescribed error tolerance, and $\delta, \delta' \in (0,1]$ designates a target failure probability.

5.1.1 Conservation of Non-Spikiness

Our first lemma establishes that the uniformly sampled submatrices of an α-spiky matrix are themselves nearly α-spiky with high probability. This property will allow for accurate submatrix completion or outlier removal using standard MC and RMF algorithms. Its proof is given in Section K.

Lemma 17 (Conservation of Non-Spikiness) *Let $\mathbf{L}_C \in \mathbb{R}^{m\times l}$ be a matrix of l columns of $\mathbf{L} \in \mathbb{R}^{m\times n}$ sampled uniformly without replacement. If $l \geq \alpha^4(\mathbf{L})\log(1/\delta)/(2\epsilon^2)$, then*

$$\alpha(\mathbf{L}_C) \leq \frac{\alpha(\mathbf{L})}{\sqrt{1-\epsilon}}$$

with probability at least $1-\delta$.

5.1.2 Column Projection Analysis

Our first theorem asserts that, with high probability, column projection produces an approximation nearly as good as a given rank-r target by sampling a number of columns proportional to the spikiness and $r\log(mn)$.

Theorem 18 (Column Projection under Non-Spikiness) *Given a matrix $\mathbf{M} \in \mathbb{R}^{m\times n}$ and a rank-r, α-spiky approximation $\mathbf{L} \in \mathbb{R}^{m\times n}$, choose*

$$l \geq 8r\alpha^4\log(2mn/\delta)/\epsilon^2,$$

and let $\mathbf{C} \in \mathbb{R}^{m\times l}$ be a matrix of l columns of $\mathbf{M}$ sampled uniformly without replacement. Then,

$$\|\mathbf{M} - \mathbf{L}^{proj}\|_F \leq \|\mathbf{M}-\mathbf{L}\|_F + \epsilon$$

with probability at least $1-\delta$, whenever $\|\mathbf{M}\|_\infty \leq \alpha/\sqrt{mn}$.

The proof of Theorem 18 builds upon the randomized matrix multiplication work of Drineas et al. (2006a,b) and will be given in Section L.

5.2 Base Algorithm Guarantee

The next result, a reformulation of Negahban and Wainwright (2012, Corollary 1), is a prototypical example of a spikiness-based estimation guarantee for noisy MC. Corollary 19 bounds the estimation error of a convex optimization approach to noisy matrix completion, under non-spikiness and uniform sampling assumptions.

Corollary 19 (Noisy MC under Non-Spikiness) (Negahban and Wainwright, 2012) *Suppose that $\mathbf{L}_0 \in \mathbb{R}^{m\times n}$ is α-spiky with rank r and $\|\mathbf{L}_0\|_F \leq 1$ and that $\mathbf{Z}_0 \in \mathbb{R}^{m\times n}$ has i.i.d. zero-mean, sub-exponential entries with variance ν^2/mn. If, for an oversampling parameter $\beta > 0$,*

$$s \geq \alpha^2\beta r(m+n)\log(m+n)$$

entries of $\mathbf{M} = \mathbf{L}_0 + \mathbf{Z}_0$ are observed with locations Ω sampled uniformly with replacement, then any solution $\hat{\mathbf{L}}$ of the problem

$$\text{minimize}_{\mathbf{L}} \quad \frac{mn}{2s}\|\mathcal{P}_\Omega(\mathbf{M}-\mathbf{L})\|_F^2 + \lambda\|\mathbf{L}\|_* \quad \text{subject to} \quad \|\mathbf{L}\|_\infty \leq \frac{\alpha}{\sqrt{mn}} \tag{3}$$
$$\text{with} \quad \lambda = 4\nu\sqrt{(m+n)\log(m+n)/s}$$

satisfies

$$\|\mathbf{L}_0 - \hat{\mathbf{L}}\|_F^2 \leq c_1\max(\nu^2, 1)/\beta$$

with probability at least $1 - c_2\exp(-c_3\log(m+n))$ for positive constants c_1, c_2, and c_3.

5.3 Spikiness Master Theorem

We now show that the same spikiness conditions that allow for accurate MC also imply high-probability estimation guarantees for DFC. To make this precise, we let $\mathbf{M} = \mathbf{L}_0 + \mathbf{Z}_0 \in \mathbb{R}^{m\times n}$, where $\mathbf{L}_0$ is α-spiky with rank r and that $\mathbf{Z}_0 \in \mathbb{R}^{m\times n}$ has i.i.d. zero-mean, sub-exponential entries with variance ν^2/mn. We further fix any $\epsilon, \delta \in (0, 1]$. Then, our Theorem 20 provides a generic bound on estimation error for DFC when used in combination with an arbitrary base algorithm. The proof, which builds upon the results of Section 5.1, is deferred to Section M.

Theorem 20 (Spikiness Master Theorem) *Choose $t = n/l$, $l \geq 13r\alpha^4 \log(4mn/\delta)/\epsilon^2$, and $p \geq 242\, r \log(14/\delta)/\epsilon^2$. Under the notation of Algorithms 1 and 2, let $\{\mathbf{C}_{0,1}, \cdots, \mathbf{C}_{0,t}\}$ be the corresponding partition of $\mathbf{L}_0$. Then, with probability at least $1-\delta$,* DFC-Proj *and* DFC-RP *guarantee that $\mathbf{C}_{0,i}$ is $(\sqrt{1.25}\alpha)$-spiky for all i and that*

$$\|\mathbf{L}_0 - \hat{\mathbf{L}}^{proj}\|_F \leq 2\sqrt{\textstyle\sum_{i=1}^t \|\mathbf{C}_{0,i} - \hat{\mathbf{C}}_i\|_F^2} + \epsilon \quad \textit{and}$$
$$\|\mathbf{L}_0 - \hat{\mathbf{L}}^{rp}\|_F \leq (2+\epsilon)\sqrt{\textstyle\sum_{i=1}^t \|\mathbf{C}_{0,i} - \hat{\mathbf{C}}_i\|_F^2}$$

whenever $\|\hat{\mathbf{C}}_i\|_\infty \leq \sqrt{1.25}\alpha/\sqrt{ml}$ for all i.

Remark 21 *The factor of $\sqrt{1.25}$ can be replaced with the smaller term $\sqrt{1+\epsilon/(4\sqrt{r})}$.*

When a target matrix is non-spiky, Theorem 20 asserts that, with high probability, the estimation error of DFC is not much larger than the error sustained by the base algorithm on each subproblem. Theorem 20 further bounds the spikiness of each submatrix with high probability, and hence we can use any spikiness-based matrix estimation guarantee to control the estimation error on each subproblem. The next section demonstrates how Theorem 20 can be applied to derive specific DFC estimation guarantees for noisy MC.

5.4 Consequences for Noisy MC

Our corollary of Theorem 20 shows that DFC retains the high-probability estimation guarantees of a standard MC solver while operating on matrices of much smaller dimension. Suppose that a base MC algorithm solves the convex optimization problem of Eq. (3). Then, Corollary 22 follows from the Spikiness Master Theorem (Theorem 20) and the base algorithm guarantee of Corollary 19.

Corollary 22 (DFC-MC under Non-Spikiness) *Suppose that $\mathbf{L}_0 \in \mathbb{R}^{m\times n}$ is α-spiky with rank r and $\|\mathbf{L}_0\|_F \leq 1$ and that $\mathbf{Z}_0 \in \mathbb{R}^{m\times n}$ has i.i.d. zero-mean, sub-exponential entries with variance ν^2/mn. Let c_1, c_2, and c_3 be positive constants as in Corollary 19. If s entries of $\mathbf{M} = \mathbf{L}_0 + \mathbf{Z}_0$ are observed with locations Ω sampled uniformly with replacement, and the base algorithm solves the optimization problem of Eq. (3), then it suffices to choose $t = n/l$,*

$$l \geq 13(c_3+1)\sqrt{\frac{(m+n)\log(m+n)\beta}{s}} nr\alpha^4 \log(4mn)/\epsilon^2,$$

and $p \geq 242\ r\log(14(m+l)^{c_3})/\epsilon^2$ *to achieve*

$$\|\mathbf{L}_0 - \hat{\mathbf{L}}^{proj}\|_F \leq 2\sqrt{c_1 \max((l/n)\nu^2, 1)/\beta} + \epsilon \quad \textit{and}$$
$$\|\mathbf{L}_0 - \hat{\mathbf{L}}^{rp}\|_F \leq (2+\epsilon)\sqrt{c_1 \max((l/n)\nu^2, 1)/\beta}$$

with respective probability at least $1-(t+1)(c_2+1)\exp(-c_3\log(m+l))$*, if the base algorithm of Eq. (3) is used with* $\lambda = 4\nu\sqrt{(m+n)\log(m+n)/s}$*.*

Remark 23 *Corollary 22 allows for the fraction of columns sampled to decrease as the number of revealed entries, s, increases. Only a vanishingly small fraction of columns* $(l/n \to 0)$ *need be sampled whenever* $s = \omega((m+n)\log^3(m+n))$*.*

To understand the conclusions of Corollary 22, consider the base algorithm of Corollary 19, which, when applied to $\mathbf{M}$, recovers an estimate $\hat{\mathbf{L}}$ satisfying

$$\|\mathbf{L}_0 - \hat{\mathbf{L}}\|_F \leq \sqrt{c_1 \max(\nu^2, 1)/\beta}$$

with high probability. Corollary 14 asserts that, with appropriately reduced probability, DFC-RP exhibits the same estimation error scaled by an adjustable factor of $2+\epsilon$, while DFC-Proj exhibits at most twice this error plus an adjustable factor of ϵ. Hence, DFC can quickly provide near-optimal estimation for non-spiky matrices as well as incoherent matrices, even when entries are missing. The proof of Corollary 22 can be found in Section N.

6. Experimental Evaluation

We now explore the accuracy and speed-up of DFC on a variety of simulated and real-world data sets. We use the Accelerated Proximal Gradient (APG) algorithm of Toh and Yun (2010) as our base noisy MC algorithm[5] and the APG algorithm of Lin et al. (2009b) as our base noisy RMF algorithm. In order to provide a fair comparison with baseline algorithms, we perform all experiments on an x86-64 architecture using a single 2.60 Ghz core and 30GB of main memory. In practice, one will typically run DFC jobs in a distributed fashion across a cluster; our released code supports this standard use case. We use the default parameter settings suggested by Toh and Yun (2010) and Lin et al. (2009b), and measure estimation error via root mean square error (RMSE). To achieve a fair running time comparison, we execute each subproblem in the F step of DFC in a serial fashion on the same machine using a single core. Since, in practice, each of these subproblems would be executed in parallel, the *parallel running time* of DFC is calculated as the time to complete the D and C steps of DFC plus the running time of the longest running subproblem in the F step. We compare DFC with two baseline methods: the base algorithm APG applied to the full matrix $\mathbf{M}$ and Partition, which carries out the D and F steps of DFC-Proj but omits the final C step (projection). We denote a particular sampling method along with the size of its partitions as '*method-xx*%,' e.g., Proj-25% refers to DFC-Proj with partitioned submatrices containing 25% of the columns of the full matrix (i.e., $t = 4$). For Partition, DFC-Proj, and DFC-RP, we orient our data matrices such that $n \geq m$ and partition by column. Moreover, for DFC-RP we set $p = 5$ and $q = 2$.

5. Our experiments with the Augmented Lagrange Multiplier (ALM) algorithm of Lin et al. (2009a) as a base algorithm (not reported) yield comparable relative speedups and performance for DFC.

6.1 Simulations

For our simulations, we focused on square matrices ($m = n$) and generated random low-rank and sparse decompositions, similar to the schemes used in related work (Candès et al., 2011; Keshavan et al., 2010; Zhou et al., 2010). We created $\mathbf{L}_0 \in \mathbb{R}^{m \times m}$ as a random product, $\mathbf{A}\mathbf{B}^\top$, where $\mathbf{A}$ and $\mathbf{B}$ are $m \times r$ matrices with independent $\mathcal{N}(0, \sqrt{1/r})$ entries such that each entry of $\mathbf{L}_0$ has unit variance. $\mathbf{Z}_0$ contained independent $\mathcal{N}(0, 0.1)$ entries. In the MC setting, s entries of $\mathbf{L}_0 + \mathbf{Z}_0$ were revealed uniformly at random. In the RMF setting, the support of $\mathbf{S}_0$ was generated uniformly at random, and the s corrupted entries took values in $[0, 1]$ with uniform probability. For each algorithm, we report error between $\mathbf{L}_0$ and the estimated low-rank matrix, and all reported results are averages over ten trials.

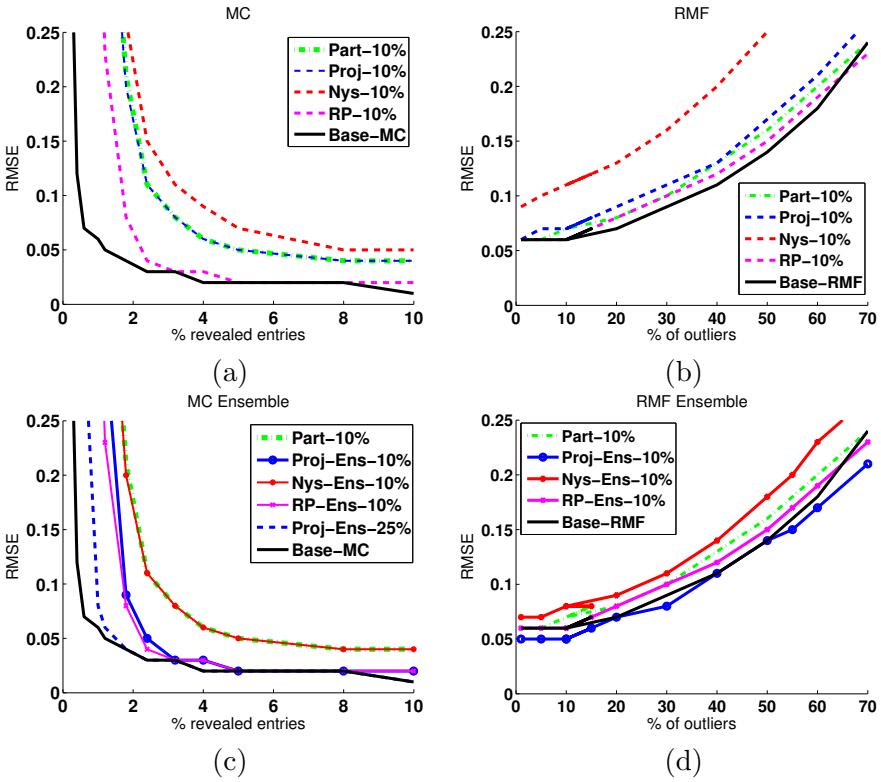


Figure 1: Recovery error of DFC relative to base algorithms.

We first explored the estimation error of DFC as a function of s, using (m = 10K, $r = 10$) with varying observation sparsity for MC and (m = 1K, $r = 10$) with a varying percentage of outliers for RMF. The results are summarized in Figure 1. In both MC and RMF, the gaps in estimation between APG and DFC are small when sampling only 10% of rows and columns. Moreover, of the standard DFC algorithms, DFC-RP performs the best, as shown in Figures 1(a) and (b). Ensembling improves the performance of DFC-Nys and DFC-Proj, as shown in Figures 1(c) and (d), and DFC-Proj-Ens in particular consistently outperforms Partition and DFC-Nys-Ens, slightly outperforms DFC-RP, and matches the performance of APG for most settings of s. In practice we observe that $\mathbf{L}^{rp}$ equals the optimal (with respect to the spectral or Frobenius norm) rank-k approximation

of $[\hat{\mathbf{C}}_1, \ldots, \hat{\mathbf{C}}_t]$, and thus the performance of DFC-RP consistently matches that of DFC-RP-Ens. We therefore omit the DFC-RP-Ens results in the remainder this section.

We next explored the speed-up of DFC as a function of matrix size. For MC, we revealed 4% of the matrix entries and set $r = 0.001 \cdot m$, while for RMF we fixed the percentage of outliers to 10% and set $r = 0.01 \cdot m$. We sampled 10% of rows and columns and observed that estimation errors were comparable to the errors presented in Figure 1 for similar settings of s; in particular, at all values of n for both MC and RMF, the errors of APG and DFC-Proj-Ens were nearly identical. Our timing results, presented in Figure 2, illustrate a near-linear speed-up for MC and a superlinear speed-up for RMF across varying matrix sizes. Note that the timing curves of the DFC algorithms and Partition all overlap, a fact that highlights the minimal computational cost of the final matrix approximation step.

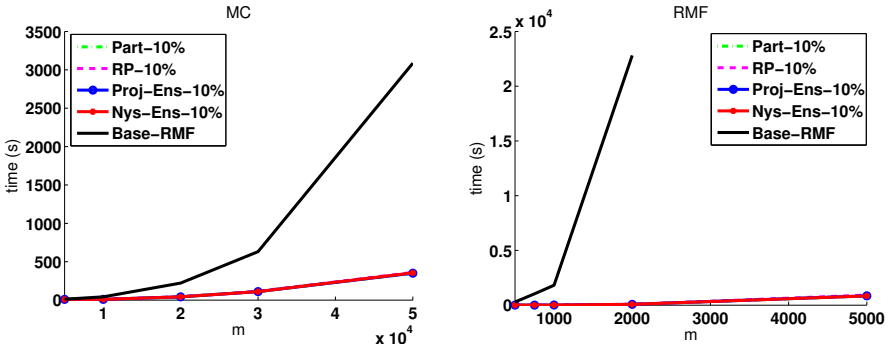


Figure 2: Speed-up of DFC relative to base algorithms.

6.2 Collaborative Filtering

Collaborative filtering for recommender systems is one prevalent real-world application of noisy matrix completion. A collaborative filtering data set can be interpreted as the incomplete observation of a ratings matrix with columns corresponding to users and rows corresponding to items. The goal is to infer the unobserved entries of this ratings matrix. We evaluate DFC on two of the largest publicly available collaborative filtering data sets: MovieLens 10M (`http://www.grouplens.org/`) with $m = 10$K, $n = 72$K, $s > 10$M, and the Netflix Prize data set (`http://www.netflixprize.com/`) with $m = 18$K, $n = 480$K, $s > 100$M. To generate test sets drawn from the training distribution, for each data set, we aggregated all available rating data into a single training set and withheld test entries uniformly at random, while ensuring that at least one training observation remained in each row and column. The algorithms were then run on the remaining training portions and evaluated on the test portions of each split. The results, averaged over three train-test splits, are summarized in Table 2. Notably, DFC-Proj, DFC-Proj-Ens, DFC-Nys-Ens, and DFC-RP all outperform Partition, and DFC-Proj-Ens performs comparably to APG while providing a nearly linear parallel time speed-up. Similar to the simulation results presented in Figure 1, DFC-RP performs the best of the standard DFC algorithms, though DFC-Proj-Ens slightly outperforms DFC-RP. Moreover, the poorer performance of DFC-Nys can be in part explained by the asymmetry of these problems. Since these matrices have many more columns than rows, MF on column submatrices is inherently

Method	MovieLens 10M		Netflix	
	RMSE	Time	RMSE	Time
Base algorithm (APG)	**0.8005**	**552.3s**	**0.8433**	**4775.4s**
Partition-25%	0.8146	146.2s	0.8451	1274.6s
Partition-10%	0.8461	56.0s	0.8491	548.0s
DFC-Nys-25%	0.8449	141.9s	0.8832	1541.2s
DFC-Nys-10%	0.8776	82.5s	0.9228	797.4s
DFC-Nys-Ens-25%	0.8085	153.5s	0.8486	1661.2s
DFC-Nys-Ens-10%	0.8328	96.2s	0.8613	909.8s
DFC-Proj-25%	0.8061	146.3s	0.8436	1274.8s
DFC-Proj-10%	0.8270	56.0s	0.8486	548.1s
DFC-Proj-Ens-25%	**0.7944**	**146.3s**	**0.8411**	**1274.8s**
DFC-Proj-Ens-10%	**0.8117**	**56.0s**	**0.8434**	**548.1s**
DFC-RP-25%	0.8027	147.4s	0.8438	1283.6s
DFC-RP-10%	0.8074	56.2s	0.8448	550.1s

Table 2: Performance of DFC relative to base algorithm APG on collaborative filtering tasks.

easier than MF on row submatrices, and for DFC-Nys, we observe that $\hat{\mathbf{C}}$ is an accurate estimate while $\hat{\mathbf{R}}$ is not.

6.3 Background Modeling in Computer Vision

Background modeling has important practical ramifications for detecting activity in surveillance video. This problem can be framed as an application of noisy RMF, where each video frame is a column of some matrix ($\mathbf{M}$), the background model is low-rank ($\mathbf{L}_0$), and moving objects and background variations, e.g., changes in illumination, are outliers ($\mathbf{S}_0$). We evaluate DFC on two videos (treating each frame as a row): 'Hall' (200 frames of size 176×144) contains significant foreground variation and was studied by Candès et al. (2011), while 'Lobby' (1546 frames of size 168×120) includes many changes in illumination (a smaller video with 250 frames was studied by Candès et al. 2011). We focused on DFC-Proj-Ens, due to its superior performance in previous experiments, and measured the RMSE between the background model estimated by DFC and that of APG. On both videos, DFC-Proj-Ens estimated nearly the same background model as the full APG algorithm in a small fraction of the time. On 'Hall,' the DFC-Proj-Ens-5% and DFC-Proj-Ens-0.5% models exhibited RMSEs of 0.564 and 1.55, quite small given pixels with 256 intensity values. The associated running time was reduced from 342.5s for APG to real-time (5.2s for a 13s video) for DFC-Proj-Ens-0.5%. Snapshots of the results are presented in Figure 3. On 'Lobby,'

Original frame | APG (342.5s) | 5% sampled (24.2s) | 0.5% sampled (5.2s)

Figure 3: Sample 'Hall' estimation by APG, DFC-Proj-Ens-5%, and DFC-Proj-Ens-.5%.

the RMSE of DFC-Proj-Ens-4% was 0.64, and the speed-up over APG was more than 20X, i.e., the running time reduced from 16557s to 792s.

6.4 From Theory to Practice

Our experimental results suggest that the theoretical error bounds of Secs. 4 and 5 can be further tightened. In particular, our master theorems Theorems 12 and 20 guarantee that DFC-Proj-Ens and DFC-RP are never more than a constant factor worse than Partition, yet in both real data experiments and simulations we observe significant gains in accuracy over Partition due to the incorporation of projection and ensembling. Moreover, our theory gives rise to comparable estimation guarantees for DFC-Nys, albeit under stronger assumptions as noted in Remark 13. This is a surprising fact given that DFC-Nys may make use of only a vanishingly small subset of all available matrix entries; however, we find that for data sets with high noise levels, methods that make use of all available data like DFC-Proj and DFC-RP are unsurprisingly more accurate than DFC-Nys. We view addressing these gaps between theory and practice as important directions for future work.

7. Conclusions

To improve the scalability of existing matrix factorization algorithms while leveraging the ubiquity of parallel computing architectures, we introduced, evaluated, and analyzed DFC, a divide-and-conquer framework for noisy matrix factorization with missing entries or outliers. DFC is trivially parallelized and particularly well suited for distributed environments given its low communication footprint. Moreover, DFC provably maintains the estimation guarantees of its base algorithm, even in the presence of noise, and yields linear to super-linear speedups in practice. A number of natural follow-up questions suggest themselves:

- Can the sampling complexities and conclusions of our theoretical analyses be strengthened? For example, can the $(2+\epsilon)$ approximation guarantees of our master theorems be sharpened to $(1+\epsilon)$? More generally, can we close the gaps between theory and practice described in Section 6.4?

- How does DFC compare empirically with scalable heuristics for MC and RMF that have little theoretical backing (see, e.g., Zhou et al., 2008; Gemulla et al., 2011; Recht and Ré, 2011; F. Niu et al., 2011; Yu et al., 2012; Mu et al., 2011)? Is improved performance obtained by pairing DFC with base algorithms lacking theoretical guarantees but displaying other practical benefits?

- Which algorithmic refinements lead to enhanced performance for DFC? For instance, could ensemble variants of DFC be improved by learning combination weights in a manner analogous to that of Kumar et al. (2009b)? In the matrix completion setting, could one use held-out entries to determine the optimal dimension (via rows or via columns) for partitioning in DFC-Proj or DFC-RP?

These open questions are fertile ground for future work.

Acknowledgments

Lester Mackey gratefully acknowledges the support of DARPA through the National Defense Science and Engineering Graduate Fellowship Program. Ameet Talwalkar gratefully acknowledges support from NSF award No. 1122732.

Appendix A. Proof of Theorem 5: Subsampled Regression under Incoherence

We now give a proof of Theorem 5. While the results of this section are stated in terms of i.i.d. with-replacement sampling of columns and rows, a concise argument due to Hoeffding (1963, Section 6) implies the same conclusions when columns and rows are sampled without replacement.

Our proof of Theorem 5 will require a strengthened version of the randomized ℓ_2 regression work of Drineas et al. (2008, Theorem 5). The proof of Theorem 5 of Drineas et al. (2008) relies heavily on the fact that $\|\mathbf{AB} - \mathbf{GH}\|_F \leq \frac{\epsilon}{2}\|\mathbf{A}\|_F\|\mathbf{B}\|_F$ with probability at least 0.9, when $\mathbf{G}$ and $\mathbf{H}$ contain sufficiently many rescaled columns and rows of $\mathbf{A}$ and $\mathbf{B}$, sampled according to a particular non-uniform probability distribution. A result of Hsu et al. (2012), modified to allow for slack in the probabilities, establishes a related claim with improved sampling complexity.[6]

Lemma 24 (Hsu et al. 2012, Example 4.3) *Given a matrix $\mathbf{A} \in \mathbb{R}^{m\times k}$ with $r \geq \operatorname{rank}(\mathbf{A})$, an error tolerance $\epsilon \in (0, 1]$, and a failure probability $\delta \in (0, 1]$, define probabilities p_j satisfying*

$$p_j \geq \frac{\beta}{Z}\|\mathbf{A}_{(j)}\|^2, \quad Z = \sum_j \|\mathbf{A}_{(j)}\|^2, \quad \text{and} \quad \textstyle\sum_{j=1}^k p_j = 1$$

6. The general conclusion of (Hsu et al., 2012, Example 4.3) is incorrectly stated as noted in Hsu (2012). However, the original statement is correct in the special case when a matrix is multiplied by its own transpose, which is the case of interest here.

for some $\beta \in (0,1]$. *Let* $\mathbf{G} \in \mathbb{R}^{m\times l}$ *be a column submatrix of* $\mathbf{A}$ *in which exactly* $l \geq 48r\log(4r/(\beta\delta))/(\beta\epsilon^2)$ *columns are selected in i.i.d. trials in which the* j*-th column is chosen with probability* p_j. *Further, let* $\mathbf{D} \in \mathbb{R}^{l\times l}$ *be a diagonal rescaling matrix with entry* $\mathbf{D}_{tt} = 1/\sqrt{lp_j}$ *whenever the* j*-th column of* $\mathbf{A}$ *is selected on the* t*-th sampling trial, for* $t = 1,\ldots,l$. *Then, with probability at least* $1-\delta$,

$$\|\mathbf{A}\mathbf{A}^\top - \mathbf{G}\mathbf{D}\mathbf{D}\mathbf{G}^\top\|_2 \leq \frac{\epsilon}{2}\|\mathbf{A}\|_2^2.$$

Using Lemma 24, we now establish a stronger version of Lemma 1 of Drineas et al. (2008). For a given $\beta \in (0,1]$ and $\mathbf{L} \in \mathbb{R}^{m\times n}$ with rank r, we first define column sampling probabilities p_j satisfying

$$p_j \geq \frac{\beta}{r}\|(\mathbf{V}_L)_{(j)}\|^2 \quad \text{and} \quad \textstyle\sum_{j=1}^n p_j = 1. \tag{4}$$

We further let $\mathbf{S} \in \mathbb{R}^{n\times l}$ be a random binary matrix with independent columns, where a single 1 appears in each column, and $\mathbf{S}_{jt} = 1$ with probability p_j for each $t \in \{1,\ldots,l\}$. Moreover, let $\mathbf{D} \in \mathbb{R}^{l\times l}$ be a diagonal rescaling matrix with entry $\mathbf{D}_{tt} = 1/\sqrt{lp_j}$ whenever $\mathbf{S}_{jt} = 1$. Postmultiplication by $\mathbf{S}$ is equivalent to selecting l random columns of a matrix, independently and with replacement. Under this notation, we establish the following lemma:

Lemma 25 *Let* $\epsilon \in (0,1]$, *and define* $\mathbf{V}_l^\top = \mathbf{V}_L^\top\mathbf{S}$ *and* $\Gamma = (\mathbf{V}_l^\top\mathbf{D})^+ - (\mathbf{V}_l^\top\mathbf{D})^\top$. *If* $l \geq 48r\log(4r/(\beta\delta))/(\beta\epsilon^2)$ *for* $\delta \in (0,1]$ *then with probability at least* $1-\delta$*:*

$$\begin{aligned}
\operatorname{rank}(\mathbf{V}_l) &= \operatorname{rank}(\mathbf{V}_L) = \operatorname{rank}(\mathbf{L}) \\
\|\Gamma\|_2 &= \|\mathbf{\Sigma}^{-1}_{V_l^\top D} - \mathbf{\Sigma}_{V_l^\top D}\|_2 \\
(\mathbf{LSD})^+ &= (\mathbf{V}_l^\top\mathbf{D})^+\mathbf{\Sigma}_L^{-1}\mathbf{U}_L^\top \\
\|\mathbf{\Sigma}^{-1}_{V_l^\top D} - \mathbf{\Sigma}_{V_l^\top D}\|_2 &\leq \epsilon/\sqrt{2}.
\end{aligned}$$

Proof By Lemma 24, for all $1 \leq i \leq r$,

$$\begin{aligned}
|1-\sigma_i^2(\mathbf{V}_l^\top\mathbf{D})| &= |\sigma_i(\mathbf{V}_L^\top\mathbf{V}_L) - \sigma_i(\mathbf{V}_l^\top\mathbf{D}\mathbf{D}\mathbf{V}_l)| \\
&\leq \|\mathbf{V}_L^\top\mathbf{V}_L - \mathbf{V}_L^\top\mathbf{S}\mathbf{D}\mathbf{D}\mathbf{S}^\top\mathbf{V}_L\|_2 \\
&\leq \epsilon/2\|\mathbf{V}_L^\top\|_2^2 = \epsilon/2,
\end{aligned}$$

where $\sigma_i(\cdot)$ is the i-th largest singular value of a given matrix. Since $\epsilon/2 \leq 1/2$, each singular value of $\mathbf{V}_l$ is positive, and so $\operatorname{rank}(\mathbf{V}_l) = \operatorname{rank}(\mathbf{V}_L) = \operatorname{rank}(\mathbf{L})$. The remainder of the proof is identical to that of Lemma 1 of Drineas et al. (2008). ■

Lemma 25 immediately yields improved sampling complexity for the randomized ℓ_2 regression of Drineas et al. (2008):

Proposition 26 *Suppose* $\mathbf{B} \in \mathbb{R}^{p\times n}$ *and* $\epsilon \in (0,1]$. *If* $l \geq 3200r\log(4r/(\beta\delta))/(\beta\epsilon^2)$ *for* $\delta \in (0,1]$, *then with probability at least* $1-\delta-0.2$*:*

$$\|\mathbf{B} - \mathbf{B}\mathbf{S}\mathbf{D}(\mathbf{L}\mathbf{S}\mathbf{D})^+\mathbf{L}\|_F \leq (1+\epsilon)\|\mathbf{B} - \mathbf{B}\mathbf{L}^+\mathbf{L}\|_F.$$

Proof The proof is identical to that of Theorem 5 of Drineas et al. (2008) once Lemma 25 is substituted for Lemma 1 of Drineas et al. (2008). ■

A typical application of Prop. 26 would involve performing a truncated SVD of $\mathbf{M}$ to obtain the *statistical leverage scores*, $\|(\mathbf{V}_L)_{(j)}\|^2$, used to compute the column sampling probabilities of Eq. (4). Here, we will take advantage of the slack term, β, allowed in the sampling probabilities of Eq. (4) to show that uniform column sampling gives rise to the same estimation guarantees for column projection approximations when $\mathbf{L}$ is sufficiently incoherent.

To prove Theorem 5, we first notice that $n \geq r\mu_0(\mathbf{V}_L)$ and hence

$$\begin{aligned} l &\geq 3200 r\mu_0(\mathbf{V}_L)\log(4r\mu_0(\mathbf{V}_L)/\delta)/\epsilon^2 \\ &\geq 3200 r\log(4r/(\beta\delta))/(\beta\epsilon^2) \end{aligned}$$

whenever $\beta \geq 1/\mu_0(\mathbf{V}_L)$. Thus, we may apply Prop. 26 with $\beta = 1/\mu_0(\mathbf{V}_L) \in (0,1]$ and $p_j = 1/n$ by noting that

$$\frac{\beta}{r}\|(\mathbf{V}_L)_{(j)}\|^2 \leq \frac{\beta}{r}\frac{r}{n}\mu_0(\mathbf{V}_L) = \frac{1}{n} = p_j$$

for all j, by the definition of $\mu_0(\mathbf{V}_L)$. By our choice of probabilities, $\mathbf{D} = \mathbf{I}\sqrt{n/l}$, and hence

$$\|\mathbf{B} - \mathbf{B}_C\mathbf{L}_C^+\mathbf{L}\|_F = \|\mathbf{B} - \mathbf{B}_C\mathbf{D}(\mathbf{L}_C\mathbf{D})^+\mathbf{L}\|_F \leq (1+\epsilon)\|\mathbf{B} - \mathbf{B}\mathbf{L}^+\mathbf{L}\|_F$$

with probability at least $1 - \delta - 0.2$, as desired.

Appendix B. Proof of Lemma 4: Conservation of Incoherence

Since for all $n > 1$,

$$c\log(n)\log(1/\delta) = (c/4)\log(n^4)\log(1/\delta) \geq 48\log(4n^2/\delta) \geq 48\log(4r\mu_0(\mathbf{V}_L)/(\delta/n))$$

as $n \geq r\mu_0(\mathbf{V}_L)$, claim *i* follows immediately from Lemma 25 with $\beta = 1/\mu_0(\mathbf{V}_L)$, $p_j = 1/n$ for all j, and $\mathbf{D} = \mathbf{I}\sqrt{n/l}$. When $\text{rank}(\mathbf{L}_C) = \text{rank}(\mathbf{L})$, Lemma 1 of Mohri and Talwalkar (2011) implies that $\mathbf{P}_{U_{L_C}} = \mathbf{P}_{U_L}$, which in turn implies claim *ii*.

To prove claim *iii* given the conclusions of Lemma 25, assume, without loss of generality, that $\mathbf{V}_l$ consists of the first l rows of $\mathbf{V}_L$. Then if $\mathbf{L}_C = \mathbf{U}_L\mathbf{\Sigma}_L\mathbf{V}_l^\top$ has $\text{rank}(\mathbf{L}_C) = \text{rank}(\mathbf{L}) = r$, the matrix $\mathbf{V}_l$ must have full column rank. Thus we can write

$$\begin{aligned} \mathbf{L}_C^+\mathbf{L}_C &= (\mathbf{U}_L\mathbf{\Sigma}_L\mathbf{V}_l^\top)^+\mathbf{U}_L\mathbf{\Sigma}_L\mathbf{V}_l^\top \\ &= (\mathbf{\Sigma}_L\mathbf{V}_l^\top)^+\mathbf{U}_L^+\mathbf{U}_L\mathbf{\Sigma}_L\mathbf{V}_l^\top \\ &= (\mathbf{\Sigma}_L\mathbf{V}_l^\top)^+\mathbf{\Sigma}_L\mathbf{V}_l^\top \\ &= (\mathbf{V}_l^\top)^+\mathbf{\Sigma}_L^+\mathbf{\Sigma}_L\mathbf{V}_l^\top \\ &= (\mathbf{V}_l^\top)^+\mathbf{V}_l^\top \\ &= \mathbf{V}_l(\mathbf{V}_l^\top\mathbf{V}_l)^{-1}\mathbf{V}_l^\top, \end{aligned}$$

where the second and third equalities follow from $\mathbf{U}_L$ having orthonormal columns, the fourth and fifth result from $\mathbf{\Sigma}_L$ having full rank and $\mathbf{V}_l$ having full column rank, and the sixth follows from $\mathbf{V}_l^\top$ having full row rank.

Now, denote the right singular vectors of $\mathbf{L}_C$ by $\mathbf{V}_{L_C} \in \mathbb{R}^{l \times r}$. Observe that $\mathbf{P}_{V_{L_C}} = \mathbf{V}_{L_C}\mathbf{V}_{L_C}^\top = \mathbf{L}_C^+\mathbf{L}_C$, and define $\mathbf{e}_{i,l}$ as the ith column of $\mathbf{I}_l$ and $\mathbf{e}_{i,n}$ as the ith column of $\mathbf{I}_n$. Then we have,

$$\begin{aligned}
\mu_0(\mathbf{V}_{L_C}) &= \frac{l}{r}\max_{1\le i\le l}\|\mathbf{P}_{V_{L_C}}\mathbf{e}_{i,l}\|^2 \\
&= \frac{l}{r}\max_{1\le i\le l}\mathbf{e}_{i,l}^\top\mathbf{L}_C^+\mathbf{L}_C\mathbf{e}_{i,l} \\
&= \frac{l}{r}\max_{1\le i\le l}\mathbf{e}_{i,l}^\top(\mathbf{V}_l^\top)^+\mathbf{V}_l^\top\mathbf{e}_{i,l} \\
&= \frac{l}{r}\max_{1\le i\le l}\mathbf{e}_{i,l}^\top\mathbf{V}_l(\mathbf{V}_l^\top\mathbf{V}_l)^{-1}\mathbf{V}_l^\top\mathbf{e}_{i,l} \\
&= \frac{l}{r}\max_{1\le i\le l}\mathbf{e}_{i,n}^\top\mathbf{V}_L(\mathbf{V}_l^\top\mathbf{V}_l)^{-1}\mathbf{V}_L^\top\mathbf{e}_{i,n},
\end{aligned}$$

where the final equality follows from $\mathbf{V}_l^\top\mathbf{e}_{i,l} = \mathbf{V}_L^\top\mathbf{e}_{i,n}$ for all $1 \le i \le l$.

Now, defining $\mathbf{Q} = \mathbf{V}_l^\top\mathbf{V}_l$ we have

$$\begin{aligned}
\mu_0(\mathbf{V}_{L_C}) &= \frac{l}{r}\max_{1\le i\le l}\mathbf{e}_{i,n}^\top\mathbf{V}_L\mathbf{Q}^{-1}\mathbf{V}_L^\top\mathbf{e}_{i,n} \\
&= \frac{l}{r}\max_{1\le i\le l}\mathrm{Tr}\left[\mathbf{e}_{i,n}^\top\mathbf{V}_L\mathbf{Q}^{-1}\mathbf{V}_L^\top\mathbf{e}_{i,n}\right] \\
&= \frac{l}{r}\max_{1\le i\le l}\mathrm{Tr}\left[\mathbf{Q}^{-1}\mathbf{V}_L^\top\mathbf{e}_{i,n}\mathbf{e}_{i,n}^\top\mathbf{V}_L\right] \\
&\le \frac{l}{r}\|\mathbf{Q}^{-1}\|_2\max_{1\le i\le l}\|\mathbf{V}_L^\top\mathbf{e}_{i,n}\mathbf{e}_{i,n}^\top\mathbf{V}_L\|_*\,,
\end{aligned}$$

by Hölder's inequality for Schatten p-norms. Since $\mathbf{V}_L^\top\mathbf{e}_{i,n}\mathbf{e}_{i,n}^\top\mathbf{V}_L$ has rank one, we can explicitly compute its trace norm as $\|\mathbf{V}_L^\top\mathbf{e}_{i,n}\|^2 = \|\mathbf{P}_{V_L}\mathbf{e}_{i,n}\|^2$. Hence,

$$\begin{aligned}
\mu_0(\mathbf{V}_{L_C}) &\le \frac{l}{r}\|\mathbf{Q}^{-1}\|_2\max_{1\le i\le l}\|\mathbf{P}_{V_L}\mathbf{e}_{i,n}\|^2 \\
&\le \frac{l}{r}\frac{r}{n}\|\mathbf{Q}^{-1}\|_2\left(\frac{n}{r}\max_{1\le i\le n}\|\mathbf{P}_{V_L}\mathbf{e}_{i,n}\|^2\right) \\
&= \frac{l}{n}\|\mathbf{Q}^{-1}\|_2\mu_0(\mathbf{V}_L)\,,
\end{aligned}$$

by the definition of μ_0-coherence. The proof of Lemma 25 established that the smallest singular value of $\frac{n}{l}\mathbf{Q} = \mathbf{V}_l^\top\mathbf{D}\mathbf{D}\mathbf{V}_l$ is lower bounded by $1 - \frac{\epsilon}{2}$ and hence $\|\mathbf{Q}^{-1}\|_2 \le \frac{n}{l(1-\epsilon/2)}$. Thus, we conclude that $\mu_0(\mathbf{V}_{L_C}) \le \mu_0(\mathbf{V}_L)/(1-\epsilon/2)$.

To prove claim *iv* under Lemma 25, we note that

$$\begin{aligned}
\mu_1(\mathbf{L}_C) &= \sqrt{\frac{ml}{r}} \max_{\substack{1\le i\le m\\ 1\le j\le l}} |\mathbf{e}_{i,m}^\top \mathbf{U}_{L_C} \mathbf{V}_{L_C}^\top \mathbf{e}_{j,l}| \\
&\le \sqrt{\frac{ml}{r}} \max_{1\le i\le m} \|\mathbf{U}_{L_C}^\top \mathbf{e}_{i,m}\| \max_{1\le j\le l} \|\mathbf{V}_{L_C}^\top \mathbf{e}_{j,l}\| \\
&= \sqrt{r}\left(\sqrt{\frac{m}{r}} \max_{1\le i\le m} \|\mathbf{P}_{U_{L_C}} \mathbf{e}_{i,m}\|\right)\left(\sqrt{\frac{l}{r}} \max_{1\le j\le l} \|\mathbf{P}_{V_{L_C}} \mathbf{e}_{j,l}\|\right) \\
&= \sqrt{r\mu_0(\mathbf{U}_{L_C})\mu_0(\mathbf{V}_{L_C})} \le \sqrt{r\mu_0(\mathbf{U}_L)\mu_0(\mathbf{V}_L)/(1-\epsilon/2)}
\end{aligned}$$

by Hölder's inequality for Schatten p-norms, the definition of μ_0-coherence, and claims *ii* and *iii*.

Appendix C. Proof of Corollary 6: Column Projection under Incoherence

Fix $c = 48000/\log(1/0.45)$, and notice that for $n > 1$,

$$48000\log(n) \ge 3200\log(n^5) \ge 3200\log(16n).$$

Hence $l \ge 3200r\mu_0(\mathbf{V}_L)\log(16n)(\log(\delta)/\log(0.45))/\epsilon^2$.

Now partition the columns of $\mathbf{C}$ into $b = \log(\delta)/\log(0.45)$ submatrices, $\mathbf{C} = [\mathbf{C}_1, \cdots, \mathbf{C}_b]$, each with $a = l/b$ columns,[7] and let $[\mathbf{L}_{C_1}, \cdots, \mathbf{L}_{C_b}]$ be the corresponding partition of $\mathbf{L}_C$. Since

$$a \ge 3200r\mu_0(\mathbf{V}_L)\log(4n/0.25)/\epsilon^2,$$

we may apply Prop. 26 independently for each i to yield

$$\|\mathbf{M} - \mathbf{C}_i\mathbf{L}_{C_i}^+\mathbf{L}\|_F \le (1+\epsilon)\|\mathbf{M} - \mathbf{M}\mathbf{L}^+\mathbf{L}\|_F \le (1+\epsilon)\|\mathbf{M} - \mathbf{L}\|_F \tag{5}$$

with probability at least 0.55, since $\mathbf{M}\mathbf{L}^+$ minimizes $\|\mathbf{M} - \mathbf{Y}\mathbf{L}\|_F$ over all $\mathbf{Y} \in \mathbb{R}^{m\times m}$.

Since each $\mathbf{C}_i = \mathbf{C}\mathbf{S}_i$ for some matrix $\mathbf{S}_i$ and $\mathbf{C}^+\mathbf{M}$ minimizes $\|\mathbf{M} - \mathbf{C}\mathbf{X}\|_F$ over all $\mathbf{X} \in \mathbb{R}^{l\times n}$, it follows that

$$\|\mathbf{M} - \mathbf{C}\mathbf{C}^+\mathbf{M}\|_F \le \|\mathbf{M} - \mathbf{C}_i\mathbf{L}_{C_i}^+\mathbf{L}\|_F,$$

for each i. Hence, if

$$\|\mathbf{M} - \mathbf{C}\mathbf{C}^+\mathbf{M}\|_F \le (1+\epsilon)\|\mathbf{M} - \mathbf{L}\|_F,$$

fails to hold, then, for each i, Eq. (5) also fails to hold. The desired conclusion therefore must hold with probability at least $1 - 0.45^b = 1 - \delta$.

7. For simplicity, we assume that b divides l evenly.

Appendix D. Proof of Corollary 7: Generalized Nyström Method under Incoherence

With $c = 48000/\log(1/0.45)$ as in Corollary 6, we notice that for $m > 1$,

$$48000\log(m) = 16000\log(m^3) \geq 16000\log(4m).$$

Therefore,

$$\begin{aligned} d &\geq 16000r\mu_0(\mathbf{U}_C)\log(4m)(\log(\delta')/\log(0.45))/\epsilon^2 \\ &\geq 3200r\mu_0(\mathbf{U}_C)\log(4m/\delta')/\epsilon^2, \end{aligned}$$

for all $m > 1$ and $\delta' \leq 0.8$. Hence, we may apply Theorem 5 and Corollary 6 in turn to obtain

$$\|\mathbf{M} - \mathbf{C}\mathbf{W}^+\mathbf{R}\|_F \leq (1+\epsilon)\|\mathbf{M} - \mathbf{C}\mathbf{C}^+\mathbf{M}\|_F \leq (1+\epsilon)^2\|\mathbf{M} - \mathbf{L}\|$$

with probability at least $(1-\delta)(1-\delta'-0.2)$ by independence.

Appendix E. Proof of Corollary 8: Noiseless Generalized Nyström Method under Incoherence

Since $\text{rank}(\mathbf{L}) = r$, $\mathbf{L}$ admits a decomposition $\mathbf{L} = \mathbf{Y}^\top\mathbf{Z}$ for some matrices $\mathbf{Y} \in \mathbb{R}^{r\times m}$ and $\mathbf{Z} \in \mathbb{R}^{r\times n}$. In particular, let $\mathbf{Y}^\top = \mathbf{U}_L\mathbf{\Sigma}_L^{\frac{1}{2}}$ and $\mathbf{Z} = \mathbf{\Sigma}_L^{\frac{1}{2}}\mathbf{V}_L^\top$. By block partitioning $\mathbf{Y}$ and $\mathbf{Z}$ as $\mathbf{Y} = \begin{bmatrix}\mathbf{Y}_1 & \mathbf{Y}_2\end{bmatrix}$ and $\mathbf{Z} = \begin{bmatrix}\mathbf{Z}_1 & \mathbf{Z}_2\end{bmatrix}$ for $\mathbf{Y}_1 \in \mathbb{R}^{r\times d}$ and $\mathbf{Z}_1 \in \mathbb{R}^{r\times l}$, we may write $\mathbf{W} = \mathbf{Y}_1^\top\mathbf{Z}_1, \mathbf{C} = \mathbf{Y}^\top\mathbf{Z}_1$, and $\mathbf{R} = \mathbf{Y}_1^\top\mathbf{Z}$. Note that we assume that the generalized Nyström approximation is generated from sampling the first l columns and the first d rows of $\mathbf{L}$, which we do without loss of generality since the rows and columns of the original low-rank matrix can always be permuted to match this assumption.

Prop. 27 shows that, like the Nyström method (Kumar et al., 2009a), the generalized Nyström method yields exact recovery of $\mathbf{L}$ whenever $\text{rank}(\mathbf{L}) = \text{rank}(\mathbf{W})$. The same result was established in Wang et al. (2009) with a different proof.

Proposition 27 *Suppose* $r = \text{rank}(\mathbf{L}) \leq \min(d, l)$ *and* $\text{rank}(\mathbf{W}) = r$. *Then* $\mathbf{L} = \mathbf{L}^{nys}$.

Proof By appealing to our factorized block decomposition, we may rewrite the generalized Nyström approximation as $\mathbf{L}^{nys} = \mathbf{C}\mathbf{W}^+\mathbf{R} = \mathbf{Y}^\top\mathbf{Z}_1(\mathbf{Y}_1^\top\mathbf{Z}_1)^+\mathbf{Y}_1^\top\mathbf{Z}$. We first note that $\text{rank}(\mathbf{W}) = r$ implies that $\text{rank}(\mathbf{Y}_1) = r$ and $\text{rank}(\mathbf{Z}_1) = r$ so that $\mathbf{Z}_1\mathbf{Z}_1^\top$ and $\mathbf{Y}_1\mathbf{Y}_1^\top$ are full-rank. Hence, $(\mathbf{Y}_1^\top\mathbf{Z}_1)^+ = \mathbf{Z}_1^\top(\mathbf{Z}_1\mathbf{Z}_1^\top)^{-1}(\mathbf{Y}_1\mathbf{Y}_1^\top)^{-1}\mathbf{Y}_1$, yielding

$$\mathbf{L}^{nys} = \mathbf{Y}^\top\mathbf{Z}_1\mathbf{Z}_1^\top(\mathbf{Z}_1\mathbf{Z}_1^\top)^{-1}(\mathbf{Y}_1\mathbf{Y}_1^\top)^{-1}\mathbf{Y}_1\mathbf{Y}_1^\top\mathbf{Z} = \mathbf{Y}^\top\mathbf{Z} = \mathbf{L}.$$

■

Prop. 27 allows us to lower bound the probability of exact recovery with the probability of randomly selecting a rank-r submatrix. As $\text{rank}(\mathbf{W}) = r$ iff both $\text{rank}(\mathbf{Y}_1) = r$ and $\text{rank}(\mathbf{Z}_1) = r$, it suffices to characterize the probability of selecting full rank submatrices of $\mathbf{Y}$ and $\mathbf{Z}$. Following the treatment of the Nyström method in Talwalkar and Rostamizadeh

(2010), we note that $\mathbf{\Sigma}_L^{-\frac{1}{2}}\mathbf{Z} = \mathbf{V}_L^\top$ and hence that $\mathbf{Z}_1^\top \mathbf{\Sigma}_{\mathbf{L}}^{-\frac{1}{2}} = \mathbf{V}_l$ where $\mathbf{V}_l \in \mathbb{R}^{l\times r}$ contains the first l components of the leading r right singular vectors of $\mathbf{L}$. It follows that $\text{rank}(\mathbf{Z}_1) = \text{rank}(\mathbf{Z}_1^\top \mathbf{\Sigma}_L^{-\frac{1}{2}}) = \text{rank}(\mathbf{V}_l)$. Similarly, $\text{rank}(\mathbf{Y}_1) = \text{rank}(\mathbf{U}_d)$ where $\mathbf{U}_d \in \mathbb{R}^{d\times r}$ contains the first d components of the leading r left singular vectors of $\mathbf{L}$. Thus, we have

$$\mathbf{P}(\text{rank}(\mathbf{Z}_1) = r) = \mathbf{P}(\text{rank}(\mathbf{V}_l) = r) \qquad \text{and} \tag{6}$$

$$\mathbf{P}(\text{rank}(\mathbf{Y}_1) = r) = \mathbf{P}(\text{rank}(\mathbf{U}_d) = r). \tag{7}$$

Next we can apply the first result of Lemma 25 to lower bound the RHSs of Eq. (6) and Eq. (7) by selecting $\epsilon = 1$, $\mathbf{S}$ such that its diagonal entries equal 1, and $\beta = \frac{1}{\mu_0(\mathbf{V}_L)}$ for the RHS of Eq. (6) and $\beta = \frac{1}{\mu_0(\mathbf{U}_L)}$ for the RHS of Eq. (7). In particular, given the lower bounds on d and l in the statement of the corollary, the RHSs are each lower bounded by $\sqrt{1-\delta}$. Furthermore, by the independence of row and column sampling and Eq. (6) and Eq. (7), we see that

$$\begin{aligned} 1-\delta &\leq \mathbf{P}(\text{rank}(\mathbf{U}_d) = r)\mathbf{P}(\text{rank}(\mathbf{V}_l) = r) \\ &= \mathbf{P}(\text{rank}(\mathbf{Y}_1) = r)\mathbf{P}(\text{rank}(\mathbf{Z}_1) = r) \\ &= \mathbf{P}(\text{rank}(\mathbf{W}) = r). \end{aligned}$$

Finally, Prop. 27 implies that

$$\mathbf{P}(\mathbf{L} = \mathbf{L}^{nys}) \geq \mathbf{P}(\text{rank}(\mathbf{W}) = r) \geq 1-\delta,$$

which proves the statement of the theorem.

Appendix F. Proof of Corollary 9: Random Projection

Our proof rests upon the following random projection guarantee of Halko et al. (2011):

Theorem 28 (Halko et al. 2011, Theorem 10.7) *Given a matrix $\mathbf{M} \in \mathbb{R}^{m\times n}$ and a rank-r approximation $\mathbf{L} \in \mathbb{R}^{m\times n}$ with $r \geq 2$, choose an oversampling parameter $p \geq 4$, where $r + p \leq \min(m, n)$. Draw an $n \times (r+p)$ standard Gaussian matrix $\mathbf{G}$, let $\mathbf{Y} = \mathbf{MG}$. For all $u, t \geq 1$,*

$$\|\mathbf{M} - \mathbf{P}_Y\mathbf{M}\|_F \leq (1 + t\sqrt{12r/p})\|\mathbf{M} - \mathbf{M}_r\|_F + ut \cdot \frac{e\sqrt{r+p}}{p+1}\|\mathbf{M} - \mathbf{M}_r\|$$

with probability at least $1 - 5t^{-p} - 2e^{-u^2/2}$.

Fix $(u, t) = (\sqrt{2\log(7/\delta)}, e)$, and note that

$$1 - 5e^{-p} - 2e^{-u^2/2} = 1 - 5e^{-p} - 2\delta/7 \geq 1 - \delta,$$

since $p \geq \log(7/\delta)$. Hence, Theorem 28 implies that

$$\begin{aligned}\|\mathbf{M} - \mathbf{P}_Y\mathbf{M}\|_F &\leq (1 + e\sqrt{12r/p})\|\mathbf{M} - \mathbf{M}_r\|_F + \frac{e^2\sqrt{2(r+p)\log(7/\delta)}}{p+1}\|\mathbf{M} - \mathbf{M}_r\|_2 \\ &\leq \left(1 + e\sqrt{12r/p} + \frac{e^2\sqrt{2(r+p)\log(7/\delta)}}{p+1}\right)\|\mathbf{M} - \mathbf{L}\|_F \\ &\leq \left(1 + e\sqrt{12r/p} + e^2\sqrt{2r\log(7/\delta)/p}\right)\|\mathbf{M} - \mathbf{L}\|_F \\ &\leq \left(1 + 11\sqrt{2r\log(7/\delta)/p}\right)\|\mathbf{M} - \mathbf{L}\|_F \leq (1+\epsilon)\|\mathbf{M} - \mathbf{L}\|_F\end{aligned}$$

with probability at least $1 - \delta$, where the second inequality follows from $\|\mathbf{M} - \mathbf{M}_r\|_2 \leq \|\mathbf{M} - \mathbf{M}_r\|_F \leq \|\mathbf{M} - \mathbf{L}\|_F$, the third follows from $\sqrt{r+p}\sqrt{p} \leq (p+1)\sqrt{r}$ for all r and p, and the final follows from our choice of $p \geq 242\ r\log(7/\delta)/\epsilon^2$.

Next, we note, as in the proof of Theorem 9.3 of Halko et al. (2011), that

$$\|\mathbf{P}_Y\mathbf{M} - \mathbf{L}^{rp}\|_F \leq \|\mathbf{P}_Y\mathbf{M} - \mathbf{P}_Y\mathbf{M}_r\|_F \leq \|\mathbf{M} - \mathbf{M}_r\|_F \leq \|\mathbf{M} - \mathbf{L}\|_F\,.$$

The first inequality holds because $\mathbf{L}^{rp}$ is by definition the best rank-r approximation to $\mathbf{P}_Y\mathbf{M}$ and $\text{rank}(\mathbf{P}_Y\mathbf{M}_r) \leq r$. The second inequality holds since

$$\|\mathbf{M} - \mathbf{M}_r\|_F = \|\mathbf{P}_Y(\mathbf{M} - \mathbf{M}_r)\|_F + \|\mathbf{P}_Y^{\perp}(\mathbf{M} - \mathbf{M}_r)\|_F\,.$$

The final inequality holds since $\mathbf{M}_r$ is the best rank-r approximation to $\mathbf{M}$ and $\text{rank}(\mathbf{L}) = r$. Moreover, by the triangle inequality,

$$\begin{aligned}\|\mathbf{M} - \mathbf{L}^{rp}\|_F &\leq \|\mathbf{M} - \mathbf{P}_Y\mathbf{M}\|_F + \|\mathbf{P}_Y\mathbf{M} - \mathbf{L}^{rp}\|_F \\ &\leq \|\mathbf{M} - \mathbf{P}_Y\mathbf{M}\|_F + \|\mathbf{M} - \mathbf{L}\|_F\,. \qquad (8)\end{aligned}$$

Combining Eq. (8) with the first statement of the corollary yields the second statement.

Appendix G. Proof of Theorem 12: Coherence Master Theorem

G.1 Proof of DFC-Proj and DFC-RP Bounds

Let $\mathbf{L}_0 = [\mathbf{C}_{0,1}, \ldots, \mathbf{C}_{0,t}]$ and $\tilde{\mathbf{L}} = [\hat{\mathbf{C}}_1, \ldots, \hat{\mathbf{C}}_t]$. Define $A(\mathbf{X})$ as the event that a matrix $\mathbf{X}$ is $(\frac{r\mu^2}{1-\epsilon/2}, r)$-coherent and K as the event $\|\tilde{\mathbf{L}} - \hat{\mathbf{L}}^{proj}\|_F \leq (1+\epsilon)\|\mathbf{L}_0 - \tilde{\mathbf{L}}\|_F$. When K holds, we have that

$$\begin{aligned}\|\mathbf{L}_0 - \hat{\mathbf{L}}^{proj}\|_F &\leq \|\mathbf{L}_0 - \tilde{\mathbf{L}}\|_F + \|\tilde{\mathbf{L}} - \hat{\mathbf{L}}^{proj}\|_F \leq (2+\epsilon)\|\mathbf{L}_0 - \tilde{\mathbf{L}}\|_F \\ &= (2+\epsilon)\sqrt{\textstyle\sum_{i=1}^t \|\mathbf{C}_{0,i} - \hat{\mathbf{C}}_i\|_F^2},\end{aligned}$$

by the triangle inequality, and hence it suffices to lower bound $\mathbf{P}(K \cap \bigcap_i A(\mathbf{C}_{0,i}))$. Our choice of l, with a factor of $\log(2/\delta)$, implies that each $A(\mathbf{C}_{0,i})$ holds with probability at least $1 - \delta/(2n)$ by Lemma 4, while K holds with probability at least $1 - \delta/2$ by Corollary 6. Hence, by the union bound,

$$\mathbf{P}(K \cap \textstyle\bigcap_i A(\mathbf{C}_{0,i})) \geq 1 - \mathbf{P}(K^c) - \sum_i \mathbf{P}(A(\mathbf{C}_{0,i})^c) \geq 1 - \delta/2 - t\delta/(2n) \geq 1 - \delta.$$

An identical proof with Corollary 9 substituted for Corollary 6 yields the random projection result.

G.2 Proof of DFC-Nys Bound

To prove the generalized Nyström result, we redefine $\tilde{\mathbf{L}}$ and write it in block notation as:

$$\tilde{\mathbf{L}} = \begin{bmatrix} \hat{\mathbf{C}}_1 & \hat{\mathbf{R}}_2 \\ \hat{\mathbf{C}}_2 & \mathbf{L}_{0,22} \end{bmatrix}, \quad \text{where} \quad \hat{\mathbf{C}} = \begin{bmatrix} \hat{\mathbf{C}}_1 \\ \hat{\mathbf{C}}_2 \end{bmatrix}, \quad \hat{\mathbf{R}} = \begin{bmatrix} \hat{\mathbf{R}}_1 & \hat{\mathbf{R}}_2 \end{bmatrix}$$

and $\mathbf{L}_{0,22} \in \mathbb{R}^{(m-d)\times(n-l)}$ is the bottom right submatrix of $\mathbf{L}_0$. We further redefine K as the event $\|\tilde{\mathbf{L}} - \hat{\mathbf{L}}^{nys}\|_F \leq (1+\epsilon)^2 \|\mathbf{L}_0 - \tilde{\mathbf{L}}\|_F$. As above,

$$\|\mathbf{L}_0 - \hat{\mathbf{L}}^{nys}\|_F \leq \|\mathbf{L}_0 - \tilde{\mathbf{L}}\|_F + \|\tilde{\mathbf{L}} - \hat{\mathbf{L}}^{nys}\|_F \leq (2 + 2\epsilon + \epsilon^2)\|\mathbf{L}_0 - \tilde{\mathbf{L}}\|_F \leq (2+3\epsilon)\|\mathbf{L}_0 - \tilde{\mathbf{L}}\|_F,$$

when K holds, by the triangle inequality. Our choices of l and

$$d \geq cl\mu_0(\hat{\mathbf{C}}) \log(m) \log(4/\delta)/\epsilon^2 \geq cr\mu \log(m) \log(1/\delta)/\epsilon^2$$

imply that $A(\mathbf{C})$ and $A(\mathbf{R})$ hold with probability at least $1 - \delta/(2n)$ and $1 - \delta/(4n)$ respectively by Lemma 4, while K holds with probability at least $(1-\delta/2)(1-\delta/4-0.2)$ by Corollary 7. Hence, by the union bound,

$$\begin{aligned} \mathbf{P}(K \cap A(\mathbf{C}) \cap A(\mathbf{R})) &\geq 1 - \mathbf{P}(K^c) - \mathbf{P}(A(\mathbf{C})^c) - \mathbf{P}(A(\mathbf{R})^c) \\ &\geq 1 - (1 - (1-\delta/2)(1-\delta/4-0.2)) - \delta/(2n) - \delta/(4n) \\ &\geq (1-\delta/2)(1-\delta/4-0.2) - 3\delta/8 \\ &\geq (1-\delta)(1-\delta-0.2) \end{aligned}$$

for all $n \geq 2$ and $\delta \leq 0.8$.

Appendix H. Proof of Corollary 14: DFC-MC under Incoherence

H.1 Proof of DFC-Proj and DFC-RP Bounds

We begin by proving the DFC-Proj bound. Let G be the event that

$$\|\mathbf{L}_0 - \hat{\mathbf{L}}^{proj}\|_F \leq (2+\epsilon) c_e \sqrt{mn}\Delta,$$

H be the event that

$$\|\mathbf{L}_0 - \hat{\mathbf{L}}^{proj}\|_F \leq (2+\epsilon)\sqrt{\textstyle\sum_{i=1}^t \|\mathbf{C}_{0,i} - \hat{\mathbf{C}}_i\|_F^2},$$

$A(\mathbf{X})$ be the event that a matrix $\mathbf{X}$ is $(\frac{r\mu^2}{1-\epsilon/2}, r)$-coherent, and, for each $i \in \{1, \ldots, t\}$, B_i be the event that $\|\mathbf{C}_{0,i} - \hat{\mathbf{C}}_i\|_F > c_e\sqrt{ml}\Delta$.

Note that, by assumption,

$$\begin{aligned} l &\geq c\mu^2 r^2 (m+n) n\beta \log^2(m+n)/(s\epsilon^2) \geq cr\mu \log(n) 2\beta \log(m+n)/\epsilon^2 \\ &\geq cr\mu \log(n)((2\beta-2)\log(\bar{n}) + \log(2))/\epsilon^2 = cr\mu \log(n) \log(2\bar{n}^{2\beta-2})/\epsilon^2. \end{aligned}$$

Hence the Coherence Master Theorem (Theorem 12) guarantees that, with probability at least $1-\bar{n}^{2-2\beta}$, H holds and the event $A(\mathbf{C}_{0,i})$ holds for each i. Since G holds whenever H holds and B_i^c holds for each i, we have

$$\begin{aligned}
\mathbf{P}(G) &\geq \mathbf{P}(H \cap \textstyle\bigcap_i B_i^c) \geq \mathbf{P}(H \cap \bigcap_i A(\mathbf{C}_{0,i}) \cap \bigcap_i B_i^c) \\
&= \mathbf{P}(H \cap \textstyle\bigcap_i A(\mathbf{C}_{0,i}))\mathbf{P}(\bigcap_i B_i^c \mid H \cap \bigcap_i A(\mathbf{C}_{0,i})) \\
&= \mathbf{P}(H \cap \textstyle\bigcap_i A(\mathbf{C}_{0,i}))(1 - \mathbf{P}(\bigcup_i B_i \mid H \cap \bigcap_i A(\mathbf{C}_{0,i}))) \\
&\geq (1-\bar{n}^{2-2\beta})(1 - \textstyle\sum_i \mathbf{P}(B_i \mid A(\mathbf{C}_{0,i}))) \\
&\geq 1 - \bar{n}^{2-2\beta} - \textstyle\sum_i \mathbf{P}(B_i \mid A(\mathbf{C}_{0,i})).
\end{aligned}$$

To prove our desired claim, it therefore suffices to show

$$\mathbf{P}(B_i \mid A(\mathbf{C}_{0,i})) \leq 4\log(\bar{n})\bar{n}^{2-2\beta} + \bar{n}^{-2\beta} \leq 5\log(\bar{n})\bar{n}^{2-2\beta}$$

for each i.

For each i, let D_i be the event that $s_i < 32\mu' r(m+l)\beta' \log^2(m+l)$, where s_i is the number of revealed entries in $\mathbf{C}_{0,i}$,

$$\mu' \triangleq \frac{\mu^2 r}{1-\epsilon/2}, \qquad \text{and} \qquad \beta' \triangleq \frac{\beta \log(\bar{n})}{\log(\max(m,l))}.$$

By Theorem 10 and our choice of β',

$$\begin{aligned}
\mathbf{P}(B_i \mid A(\mathbf{C}_{0,i})) &\leq \mathbf{P}(B_i \mid A(\mathbf{C}_{0,i}), D_i^c) + \mathbf{P}(D_i \mid A(\mathbf{C}_{0,i})) \\
&\leq 4\log(\max(m,l))\max(m,l)^{2-2\beta'} + \mathbf{P}(D_i) \\
&\leq 4\log(\bar{n})\bar{n}^{2-2\beta} + \mathbf{P}(D_i).
\end{aligned}$$

Further, since the support of $\mathbf{S}_0$ is uniformly distributed and of cardinality s, the variable s_i has a hypergeometric distribution with $\mathbf{E}(s_i) = \frac{sl}{n}$ and hence satisfies Hoeffding's inequality for the hypergeometric distribution (Hoeffding, 1963, Section 6):

$$\mathbf{P}(s_i \leq \mathbf{E}(s_i) - st) \leq \exp\bigl(-2st^2\bigr).$$

Since, by assumption,

$$s \geq c\mu^2 r^2 (m+n) n\beta \log^2(m+n)/(l\epsilon^2) \geq 64\mu' r(m+l)n\beta' \log^2(m+l)/l,$$

and

$$sl^2/n^2 \geq c\mu^2 r^2 (m+n) l\beta \log^2(m+n)/(n\epsilon^2) \geq 4\log(\bar{n})\beta,$$

it follows that

$$\begin{aligned}
\mathbf{P}(D_i) &= \mathbf{P}\left(s_i < \mathbf{E}(s_i) - s\left(\frac{l}{n} - \frac{32\mu' r(m+l)\beta' \log^2(m+l)}{s}\right)\right) \\
&\leq \mathbf{P}\left(s_i < \mathbf{E}(s_i) - s\left(\frac{l}{n} - \frac{l}{2n}\right)\right) = \mathbf{P}\left(s_i < \mathbf{E}(s_i) - s\frac{l}{2n}\right) \\
&\leq \exp\left(-\frac{sl^2}{2n^2}\right) \leq \exp(-2\log(\bar{n})\beta) = \bar{n}^{-2\beta}.
\end{aligned}$$

Hence, $\mathbf{P}(B_i \mid A(\mathbf{C}_{0,i})) \leq 4\log(\bar{n})\bar{n}^{2-2\beta} + \bar{n}^{-2\beta}$ for each i, and the DFC-Proj result follows.

Since, $p \geq 242\ r\log(14\bar{n}^{2\beta-2})/\epsilon^2$, the DFC-RP bound follows in an identical manner from the Coherence Master Theorem (Theorem 12).

H.2 Proof of DFC-Nys Bound

For DFC-Nys, let B_C be the event that $\|\mathbf{C}_0 - \hat{\mathbf{C}}\|_F > c_e\sqrt{ml}\Delta$ and B_R be the event that $\|\mathbf{R}_0 - \hat{\mathbf{R}}\|_F > c_e\sqrt{dn}\Delta$. The Coherence Master Theorem (Theorem 12) and our choice of

$$d \geq cl\mu_0(\hat{\mathbf{C}})(2\beta-1)\log^2(4\bar{n})\bar{n}/(n\epsilon^2) \geq cl\mu_0(\hat{\mathbf{C}})\log(m)\log(4\bar{n}^{2\beta-2})/\epsilon^2$$

guarantee that, with probability at least $(1-\bar{n}^{2-2\beta})(1-\bar{n}^{2-2\beta}-0.2) \geq 1-2\bar{n}^{2-2\beta}-0.2$,

$$\|\mathbf{L}_0 - \hat{\mathbf{L}}^{nys}\|_F \leq (2+3\epsilon)\sqrt{\|\mathbf{C}_0-\hat{\mathbf{C}}\|_F^2 + \|\mathbf{R}_0-\hat{\mathbf{R}}\|_F^2},$$

and both $A(\mathbf{C})$ and $A(\mathbf{R})$ hold. Moreover, since

$$d \geq cl\mu_0(\hat{\mathbf{C}})(2\beta-1)\log^2(4\bar{n})\bar{n}/(n\epsilon^2) \geq c\mu^2r^2(m+n)\bar{n}\beta\log^2(m+n)/(s\epsilon^2),$$

reasoning identical to the DFC-Proj case yields $\mathbf{P}(B_C \mid A(\mathbf{C})) \leq 4\log(\bar{n})\bar{n}^{2-2\beta}+\bar{n}^{-2\beta}$ and $\mathbf{P}(B_R \mid A(\mathbf{R})) \leq 4\log(\bar{n})\bar{n}^{2-2\beta}+\bar{n}^{-2\beta}$, and the DFC-Nys bound follows as above.

Appendix I. Proof of Corollary 16: DFC-RMF under Incoherence

I.1 Proof of DFC-Proj and DFC-RP Bounds

We begin by proving the DFC-Proj bound. Let G be the event that

$$\|\mathbf{L}_0 - \hat{\mathbf{L}}^{proj}\|_F \leq (2+\epsilon)c_e'\sqrt{mn}\Delta$$

for the constant c_e' defined in Theorem 11, H be the event that

$$\|\mathbf{L}_0 - \hat{\mathbf{L}}^{proj}\|_F \leq (2+\epsilon)\sqrt{\textstyle\sum_{i=1}^t \|\mathbf{C}_{0,i}-\hat{\mathbf{C}}_i\|_F^2},$$

$A(\mathbf{X})$ be the event that a matrix $\mathbf{X}$ is $(\frac{r\mu^2}{1-\epsilon/2}, r)$-coherent, and, for each $i \in \{1,\dots,t\}$, B_i be the event that $\|\mathbf{C}_{0,i}-\hat{\mathbf{C}}_i\|_F > c_e'\sqrt{ml}\Delta$.

We may take $\rho_r \leq 1$, and hence, by assumption,

$$l \geq cr^2\mu^2\beta\log^2(2\bar{n})/(\epsilon^2\rho_r) \geq cr\mu\log(n)\log(2\bar{n}^\beta)/\epsilon^2.$$

Hence the Coherence Master Theorem (Theorem 12) guarantees that, with probability at least $1-\bar{n}^{-\beta}$, H holds and the event $A(\mathbf{C}_{0,i})$ holds for each i. Since G holds whenever H holds and B_i^c holds for each i, we have

$$\begin{aligned}
\mathbf{P}(G) &\geq \mathbf{P}(H \cap \textstyle\bigcap_i B_i^c) \geq \mathbf{P}(H \cap \bigcap_i A(\mathbf{C}_{0,i}) \cap \bigcap_i B_i^c) \\
&= \mathbf{P}(H \cap \textstyle\bigcap_i A(\mathbf{C}_{0,i}))\mathbf{P}(\bigcap_i B_i^c \mid H \cap \bigcap_i A(\mathbf{C}_{0,i})) \\
&= \mathbf{P}(H \cap \textstyle\bigcap_i A(\mathbf{C}_{0,i}))(1-\mathbf{P}(\bigcup_i B_i \mid H \cap \bigcap_i A(\mathbf{C}_{0,i}))) \\
&\geq (1-\bar{n}^{-\beta})(1-\textstyle\sum_i \mathbf{P}(B_i \mid A(\mathbf{C}_{0,i}))) \\
&\geq 1-\bar{n}^{-\beta}-\textstyle\sum_i \mathbf{P}(B_i \mid A(\mathbf{C}_{0,i})).
\end{aligned}$$

To prove our desired claim, it therefore suffices to show

$$\mathbf{P}(B_i \mid A(\mathbf{C}_{0,i})) \leq (c_p+1)\bar{n}^{-\beta}$$

for each i.

Define $\bar{m} \triangleq \max(m, l)$ and $\beta'' \triangleq \beta \log(\bar{n})/\log(\bar{m}) \leq \beta'$. By assumption,

$$r \leq \frac{\rho_r m}{2\mu^2 r \log^2(\bar{n})} \leq \frac{\rho_r m(1-\epsilon/2)}{\mu^2 r \log^2(\bar{m})} \quad \text{and} \quad r \leq \frac{\rho_r l \epsilon^2}{c\mu^2 r \beta \log^2(2\bar{n})} \leq \frac{\rho_r l(1-\epsilon/2)}{\mu^2 r \log^2(\bar{m})}.$$

Hence, by Theorem 11 and the definitions of β' and β'',

$$\begin{aligned}
\mathbf{P}(B_i \mid A(\mathbf{C}_{0,i})) &\leq \mathbf{P}\big(B_i \mid A(\mathbf{C}_{0,i}), s_i \leq (1-\rho_s\beta'')ml\big) + \mathbf{P}\big(s_i > (1-\rho_s\beta'')ml \mid A(\mathbf{C}_{0,i})\big) \\
&\leq c_p \bar{n}^{-\beta''} + \mathbf{P}\big(s_i > (1-\rho_s\beta'')ml\big) \\
&\leq c_p \bar{n}^{-\beta} + \mathbf{P}\big(s_i > (1-\rho_s\beta')ml\big),
\end{aligned}$$

where s_i is the number of corrupted entries in $\mathbf{C}_{0,i}$. Further, since the support of $\mathbf{S}_0$ is uniformly distributed and of cardinality s, the variable s_i has a hypergeometric distribution with $\mathbf{E}(s_i) = \frac{sl}{n}$ and hence satisfies Bernstein's inequality for the hypergeometric (Hoeffding, 1963, Section 6):

$$\mathbf{P}(s_i \geq \mathbf{E}(s_i) + st) \leq \exp\big(-st^2/(2\sigma^2 + 2t/3)\big) \leq \exp\big(-st^2 n/4l\big),$$

for all $0 \leq t \leq 3l/n$ and $\sigma^2 \triangleq \frac{l}{n}(1-\frac{l}{n}) \leq \frac{l}{n}$. It therefore follows that

$$\begin{aligned}
\mathbf{P}\big(s_i > (1-\rho_s\beta')ml\big) &= \mathbf{P}\left(s_i > \mathbf{E}(s_i) + s\left(\frac{(1-\rho_s\beta')ml}{s} - \frac{l}{n}\right)\right) \\
&= \mathbf{P}\left(s_i > \mathbf{E}(s_i) + s\frac{l}{n}\left(\frac{(1-\rho_s\beta')}{(1-\rho_s\beta_s)} - 1\right)\right) \\
&\leq \exp\left(-s\frac{l}{4n}\left(\frac{(1-\rho_s\beta')}{(1-\rho_s\beta_s)} - 1\right)^2\right) \\
&= \exp\left(-\frac{ml}{4}\frac{(\rho_s\beta_s - \rho_s\beta')^2}{(1-\rho_s\beta_s)}\right) \leq \bar{n}^{-\beta}
\end{aligned}$$

by our assumptions on s and l and the fact that $\frac{l}{n}\left(\frac{(1-\rho_s\beta')}{(1-\rho_s\beta_s)} - 1\right) \leq 3l/n$ whenever $4\beta_s - 3/\rho_s \leq \beta'$. Hence, $\mathbf{P}(B_i \mid A(\mathbf{C}_{0,i})) \leq (c_p+1)\bar{n}^{-\beta}$ for each i, and the DFC-Proj result follows.

Since, $p \geq 242\, r\log(14\bar{n}^{\beta})/\epsilon^2$, the DFC-RP bound follows in an identical manner from the Coherence Master Theorem (Theorem 12).

I.2 Proof of DFC-Nys Bound

For DFC-Nys, let B_C be the event that $\|\mathbf{C}_0 - \hat{\mathbf{C}}\|_F > c'_e\sqrt{ml}\Delta$ and B_R be the event that $\|\mathbf{R}_0 - \hat{\mathbf{R}}\|_F > c'_e\sqrt{dn}\Delta$. The Coherence Master Theorem (Theorem 12) and our choice of $d \geq cl\mu_0(\hat{\mathbf{C}})\beta\log^2(4\bar{n})/\epsilon^2$ guarantee that, with probability at least $(1-\bar{n}^{-\beta})(1-\bar{n}^{-\beta}-0.2) \geq 1 - 2\bar{n}^{-\beta} - 0.2$,

$$\|\mathbf{L}_0 - \hat{\mathbf{L}}^{nys}\|_F \leq (2+3\epsilon)\sqrt{\|\mathbf{C}_0 - \hat{\mathbf{C}}\|_F^2 + \|\mathbf{R}_0 - \hat{\mathbf{R}}\|_F^2},$$

and both $A(\mathbf{C})$ and $A(\mathbf{R})$ hold. Moreover, since

$$d \geq cl\mu_0(\hat{\mathbf{C}})\beta \log^2(4\bar{n})/\epsilon^2 \geq c\mu^2 r^2 \beta \log^2(\bar{n})/(\epsilon^2 \rho_r),$$

reasoning identical to the DFC-Proj case yields

$$\mathbf{P}(B_C \mid A(\mathbf{C})) \leq (c_p + 1)\bar{n}^{-\beta} \quad \text{and} \quad \mathbf{P}(B_R \mid A(\mathbf{R})) \leq (c_p + 1)\bar{n}^{-\beta},$$

and the DFC-Nys bound follows as above.

Appendix J. Proof of Theorem 10: Noisy MC under Incoherence

In the spirit of Candès and Plan (2010), our proof will extend the noiseless analysis of Recht (2011) to the noisy matrix completion setting. As suggested in Gross and Nesme (2010), we will obtain strengthened results, even in the noiseless case, by reasoning directly about the without-replacement sampling model, rather than appealing to a with-replacement surrogate, as done in Recht (2011).

For $\mathbf{U}_{L_0}\mathbf{\Sigma}_{L_0}\mathbf{V}_{L_0}^\top$ the compact SVD of $\mathbf{L}_0$, we let $T = \{\mathbf{U}_{L_0}\mathbf{X} + \mathbf{Y}\mathbf{V}_{L_0}^\top : \mathbf{X} \in \mathbb{R}^{r\times n}, \mathbf{Y} \in \mathbb{R}^{m\times r}\}$, $\mathcal{P}_T$ denote orthogonal projection onto the space T, and $\mathcal{P}_{T^\perp}$ represent orthogonal projection onto the orthogonal complement of T. We further define $\mathcal{I}$ as the identity operator on $\mathbb{R}^{m\times n}$ and the spectral norm of an operator $\mathcal{A} : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ as $\|\mathcal{A}\|_2 = \sup_{\|\mathbf{X}\|_F \leq 1} \|\mathcal{A}(\mathbf{X})\|_F$.

We begin with a theorem providing sufficient conditions for our desired estimation guarantee.

Theorem 29 *Under the assumptions of Theorem 10, suppose that*

$$\frac{mn}{s}\left\|\mathcal{P}_T\mathcal{P}_\Omega\mathcal{P}_T - \frac{s}{mn}\mathcal{P}_T\right\|_2 \leq \frac{1}{2} \tag{9}$$

and that there exists a $\mathbf{Y} = \mathcal{P}_\Omega(\mathbf{Y}) \in \mathbb{R}^{m\times n}$ satisfying

$$\|\mathcal{P}_T(\mathbf{Y}) - \mathbf{U}_{L_0}\mathbf{V}_{L_0}^\top\|_F \leq \sqrt{\frac{s}{32mn}} \quad \textit{and} \quad \|\mathcal{P}_{T^\perp}(\mathbf{Y})\|_2 < \frac{1}{2}. \tag{10}$$

Then,

$$\|\mathbf{L}_0 - \hat{\mathbf{L}}\|_F \leq 8\sqrt{\frac{2m^2n}{s} + m + \frac{1}{16}}\Delta \leq c_e''\sqrt{mn}\Delta.$$

Proof We may write $\hat{\mathbf{L}}$ as $\mathbf{L}_0 + \mathbf{G} + \mathbf{H}$, where $\mathcal{P}_\Omega(\mathbf{G}) = \mathbf{G}$ and $\mathcal{P}_\Omega(\mathbf{H}) = \mathbf{0}$. Then, under Eq. (9),

$$\|\mathcal{P}_\Omega\mathcal{P}_T(\mathbf{H})\|_F^2 = \left\langle \mathbf{H}, \mathcal{P}_T\mathcal{P}_\Omega^2\mathcal{P}_T(\mathbf{H})\right\rangle \geq \langle \mathbf{H}, \mathcal{P}_T\mathcal{P}_\Omega\mathcal{P}_T(\mathbf{H})\rangle \geq \frac{s}{2mn}\|\mathcal{P}_T(\mathbf{H})\|_F^2.$$

Furthermore, by the triangle inequality, $0 = \|\mathcal{P}_\Omega(\mathbf{H})\|_F \geq \|\mathcal{P}_\Omega\mathcal{P}_T(\mathbf{H})\|_F - \|\mathcal{P}_\Omega\mathcal{P}_{T^\perp}(\mathbf{H})\|_F$. Hence, we have

$$\sqrt{\frac{s}{2mn}}\|\mathcal{P}_T(\mathbf{H})\|_F \leq \|\mathcal{P}_\Omega\mathcal{P}_T(\mathbf{H})\|_F \leq \|\mathcal{P}_\Omega\mathcal{P}_{T^\perp}(\mathbf{H})\|_F \leq \|\mathcal{P}_{T^\perp}(\mathbf{H})\|_F \leq \|\mathcal{P}_{T^\perp}(\mathbf{H})\|_*, \tag{11}$$

where the penultimate inequality follows as $\mathcal{P}_\Omega$ is an orthogonal projection operator.

Next we select $\mathbf{U}_\perp$ and $\mathbf{V}_\perp$ such that $[\mathbf{U}_{L_0}, \mathbf{U}_\perp]$ and $[\mathbf{V}_{L_0}, \mathbf{V}_\perp]$ are orthonormal and $\left\langle \mathbf{U}_\perp \mathbf{V}_\perp^\top, \mathcal{P}_{T^\perp}(\mathbf{H}) \right\rangle = \|\mathcal{P}_{T^\perp}(\mathbf{H})\|_*$ and note that

$$\begin{aligned}
\|\mathbf{L}_0 + \mathbf{H}\|_* &\geq \left\langle \mathbf{U}_{L_0}\mathbf{V}_{L_0}^\top + \mathbf{U}_\perp \mathbf{V}_\perp^\top, \mathbf{L}_0 + \mathbf{H} \right\rangle \\
&= \|\mathbf{L}_0\|_* + \left\langle \mathbf{U}_{L_0}\mathbf{V}_{L_0}^\top + \mathbf{U}_\perp \mathbf{V}_\perp^\top - \mathbf{Y}, \mathbf{H} \right\rangle \\
&= \|\mathbf{L}_0\|_* + \left\langle \mathbf{U}_{L_0}\mathbf{V}_{L_0}^\top - \mathcal{P}_T(\mathbf{Y}), \mathcal{P}_T(\mathbf{H}) \right\rangle + \left\langle \mathbf{U}_\perp \mathbf{V}_\perp^\top, \mathcal{P}_{T^\perp}(\mathbf{H}) \right\rangle - \langle \mathcal{P}_{T^\perp}(\mathbf{Y}), \mathcal{P}_{T^\perp}(\mathbf{H}) \rangle \\
&\geq \|\mathbf{L}_0\|_* - \|\mathbf{U}_{L_0}\mathbf{V}_{L_0}^\top - \mathcal{P}_T(\mathbf{Y})\|_F \|\mathcal{P}_T(\mathbf{H})\|_F + \|\mathcal{P}_{T^\perp}(\mathbf{H})\|_* - \|\mathcal{P}_{T^\perp}(\mathbf{Y})\|_2 \|\mathcal{P}_{T^\perp}(\mathbf{H})\|_* \\
&> \|\mathbf{L}_0\|_* + \frac{1}{2}\|\mathcal{P}_{T^\perp}(\mathbf{H})\|_* - \sqrt{\frac{s}{32mn}}\|\mathcal{P}_T(\mathbf{H})\|_F \\
&\geq \|\mathbf{L}_0\|_* + \frac{1}{4}\|\mathcal{P}_{T^\perp}(\mathbf{H})\|_F
\end{aligned}$$

where the first inequality follows from the variational representation of the trace norm, $\|\mathbf{A}\|_* = \sup_{\|\mathbf{B}\|_2 \leq 1} \langle \mathbf{A}, \mathbf{B} \rangle$, the first equality follows from the fact that $\langle \mathbf{Y}, \mathbf{H} \rangle = 0$ for $\mathbf{Y} = \mathcal{P}_\Omega(\mathbf{Y})$, the second inequality follows from Hölder's inequality for Schatten p-norms, the third inequality follows from Eq. (10), and the final inequality follows from Eq. (11).

Since $\mathbf{L}_0$ is feasible for Eq. (1), $\|\mathbf{L}_0\|_* \geq \|\hat{\mathbf{L}}\|_*$, and, by the triangle inequality, $\|\hat{\mathbf{L}}\|_* \geq \|\mathbf{L}_0 + \mathbf{H}\|_* - \|\mathbf{G}\|_*$. Since $\|\mathbf{G}\|_* \leq \sqrt{m}\|\mathbf{G}\|_F$ and $\|\mathbf{G}\|_F \leq \|\mathcal{P}_\Omega(\hat{\mathbf{L}} - \mathbf{M})\|_F + \|\mathcal{P}_\Omega(\mathbf{M} - \mathbf{L}_0)\|_F \leq 2\Delta$, we conclude that

$$\begin{aligned}
\|\mathbf{L}_0 - \hat{\mathbf{L}}\|_F^2 &= \|\mathcal{P}_T(\mathbf{H})\|_F^2 + \|\mathcal{P}_{T^\perp}(\mathbf{H})\|_F^2 + \|\mathbf{G}\|_F^2 \\
&\leq \left(\frac{2mn}{s} + 1\right)\|\mathcal{P}_{T^\perp}(\mathbf{H})\|_F^2 + \|\mathbf{G}\|_F^2 \\
&\leq 16\left(\frac{2mn}{s} + 1\right)\|\mathbf{G}\|_*^2 + \|\mathbf{G}\|_F^2 \\
&\leq 64\left(\frac{2m^2 n}{s} + m + \frac{1}{16}\right)\Delta^2.
\end{aligned}$$

Hence

$$\|\mathbf{L}_0 - \hat{\mathbf{L}}\|_F \leq 8\sqrt{\frac{2m^2 n}{s} + m + \frac{1}{16}}\Delta \leq c_e'' \sqrt{mn}\Delta$$

for some constant c_e'', by our assumption on s. ■

To show that the sufficient conditions of Theorem 29 hold with high probability, we will require four lemmas. The first establishes that the operator $\mathcal{P}_T \mathcal{P}_\Omega \mathcal{P}_T$ is nearly an isometry on T when sufficiently many entries are sampled.

Lemma 30 *For all $\beta > 1$,*

$$\frac{mn}{s}\left\|\mathcal{P}_T \mathcal{P}_\Omega \mathcal{P}_T - \frac{s}{mn}\mathcal{P}_T\right\|_2 \leq \sqrt{\frac{16\mu r(m+n)\beta \log(n)}{3s}}$$

with probability at least $1 - 2n^{2-2\beta}$ provided that $s > \frac{16}{3}\mu r(n+m)\beta \log(n)$.

The second states that a sparsely but uniformly observed matrix is close to a multiple of the original matrix under the spectral norm.

Lemma 31 *Let* $\mathbf{Z}$ *be a fixed matrix in* $\mathbb{R}^{m\times n}$. *Then for all* $\beta > 1$,

$$\left\| \left(\frac{mn}{s} \mathcal{P}_\Omega - \mathcal{I} \right)(\mathbf{Z}) \right\|_2 \leq \sqrt{\frac{8\beta mn^2 \log(m+n)}{3s}} \|\mathbf{Z}\|_\infty$$

with probability at least $1 - (m+n)^{1-\beta}$ *provided that* $s > 6\beta m \log(m+n)$.

The third asserts that the matrix infinity norm of a matrix in T does not increase under the operator $\mathcal{P}_T \mathcal{P}_\Omega$.

Lemma 32 *Let* $\mathbf{Z} \in T$ *be a fixed matrix. Then for all* $\beta > 2$

$$\left\| \frac{mn}{s} \mathcal{P}_T \mathcal{P}_\Omega(\mathbf{Z}) - \mathbf{Z} \right\|_\infty \leq \sqrt{\frac{8\beta\mu r(m+n)\log(n)}{3s}} \|\mathbf{Z}\|_\infty$$

with probability at least $1 - 2n^{2-\beta}$ *provided that* $s > \frac{8}{3}\beta\mu r(m+n)\log(n)$.

These three lemmas were proved in Recht (2011, Theorem 6, Theorem 7, and Lemma 8) under the assumption that entry locations in Ω were sampled *with* replacement. They admit identical proofs under the sampling without replacement model by noting that the referenced Noncommutative Bernstein Inequality (Recht, 2011, Theorem 4) also holds under sampling without replacement, as shown in Gross and Nesme (2010).

Lemma 30 guarantees that Eq. (9) holds with high probability. To construct a matrix $\mathbf{Y} = \mathcal{P}_\Omega(\mathbf{Y})$ satisfying Eq. (10), we consider a sampling with batch replacement scheme recommended in Gross and Nesme (2010) and developed in Chen et al. (2011). Let $\tilde{\Omega}_1, \ldots, \tilde{\Omega}_p$ be independent sets, each consisting of q random entry locations sampled without replacement, where $pq = s$. Let $\tilde{\Omega} = \cup_{i=1}^p \tilde{\Omega}_i$, and note that there exist p and q satisfying

$$q \geq \frac{128}{3}\mu r(m+n)\beta\log(m+n) \quad \text{and} \quad p \geq \frac{3}{4}\log(n/2).$$

It suffices to establish Eq. (10) under this batch replacement scheme, as shown in the next lemma.

Lemma 33 *For any location set* $\Omega_0 \subset \{1,\ldots,m\} \times \{1,\ldots,n\}$, *let* $A(\Omega_0)$ *be the event that there exists* $\mathbf{Y} = \mathcal{P}_{\Omega_0}(\mathbf{Y}) \in \mathbb{R}^{m\times n}$ *satisfying Eq. (10). If* $\Omega(s)$ *consists of* s *locations sampled uniformly without replacement and* $\tilde{\Omega}(s)$ *is sampled via batch replacement with* p *batches of size* q *for* $pq = s$, *then* $\mathbf{P}(A(\tilde{\Omega}(s))) \leq \mathbf{P}(A(\Omega(s)))$.

Proof As sketched in Gross and Nesme (2010)

$$\begin{aligned}
\mathbf{P}\Big(A(\tilde{\Omega(s)})\Big) &= \sum_{i=1}^s \mathbf{P}(|\tilde{\Omega}| = i)\mathbf{P}(A(\tilde{\Omega}(i)) \mid |\tilde{\Omega}| = i) \\
&\leq \sum_{i=1}^s \mathbf{P}(|\tilde{\Omega}| = i)\mathbf{P}(A(\Omega(i))) \\
&\leq \sum_{i=1}^s \mathbf{P}(|\tilde{\Omega}| = i)\mathbf{P}(A(\Omega(s))) = \mathbf{P}(A(\Omega(s))),
\end{aligned}$$

since the probability of existence never decreases with more entries sampled without replacement and, given the size of $\tilde{\Omega}$, the locations of $\tilde{\Omega}$ are conditionally distributed uniformly (without replacement). ■

We now follow the construction of Recht (2011) to obtain $\mathbf{Y} = \mathcal{P}_{\tilde{\Omega}}(\mathbf{Y})$ satisfying Eq. (10). Let $\mathbf{W}_0 = \mathbf{U}_{L_0}\mathbf{V}_{L_0}^\top$ and define $\mathbf{Y}_k = \frac{mn}{q}\sum_{j=1}^{k}\mathcal{P}_{\tilde{\Omega}_j}(\mathbf{W}_{j-1})$ and $\mathbf{W}_k = \mathbf{U}_{L_0}\mathbf{V}_{L_0}^\top - \mathcal{P}_T(\mathbf{Y}_k)$ for $k = 1, \ldots, p$. Assume that

$$\frac{mn}{q}\left\|\mathcal{P}_T\mathcal{P}_{\tilde{\Omega}_k}\mathcal{P}_T - \frac{q}{mn}\mathcal{P}_T\right\|_2 \leq \frac{1}{2} \tag{12}$$

for all k. Then

$$\|\mathbf{W}_k\|_F = \left\|\mathbf{W}_{k-1} - \frac{mn}{q}\mathcal{P}_T\mathcal{P}_{\tilde{\Omega}_k}(\mathbf{W}_{k-1})\right\|_F = \left\|(\mathcal{P}_T - \frac{mn}{q}\mathcal{P}_T\mathcal{P}_{\tilde{\Omega}_k}\mathcal{P}_T)(\mathbf{W}_{k-1})\right\|_F \leq \frac{1}{2}\|\mathbf{W}_{k-1}\|_F$$

and hence $\|\mathbf{W}_k\|_F \leq 2^{-k}\|\mathbf{W}_0\|_F = 2^{-k}\sqrt{r}$. Since

$$p \geq \frac{3}{4}\log(n/2) \geq \frac{1}{2}\log_2(n/2) \geq \log_2\sqrt{32rmn/s}\,,$$

$\mathbf{Y} \triangleq \mathbf{Y}_p$ satisfies the first condition of Eq. (10).

The second condition of Eq. (10) follows from the assumptions

$$\left\|\mathbf{W}_{k-1} - \frac{mn}{q}\mathcal{P}_T\mathcal{P}_{\tilde{\Omega}_k}(\mathbf{W}_{k-1})\right\|_\infty \leq \frac{1}{2}\|\mathbf{W}_{k-1}\|_\infty \tag{13}$$

$$\left\|\left(\frac{mn}{q}\mathcal{P}_{\tilde{\Omega}_k} - \mathcal{I}\right)(\mathbf{W}_{k-1})\right\|_2 \leq \sqrt{\frac{8mn^2\beta\log(m+n)}{3q}}\|\mathbf{W}_{k-1}\|_\infty \tag{14}$$

for all k, since Eq. (13) implies $\|\mathbf{W}_k\|_\infty \leq 2^{-k}\|\mathbf{U}_{L_0}\mathbf{V}_{L_0}^\top\|_\infty$, and thus

$$\begin{aligned}
\|\mathcal{P}_{T^\perp}(\mathbf{Y}_p)\|_2 &\leq \sum_{j=1}^{p}\left\|\frac{mn}{q}\mathcal{P}_{T^\perp}\mathcal{P}_{\tilde{\Omega}_j}(\mathbf{W}_{j-1})\right\|_2 = \sum_{j=1}^{p}\left\|\mathcal{P}_{T^\perp}(\frac{mn}{q}\mathcal{P}_{\tilde{\Omega}_j}(\mathbf{W}_{j-1}) - \mathbf{W}_{j-1})\right\|_2 \\
&\leq \sum_{j=1}^{p}\left\|(\frac{mn}{q}\mathcal{P}_{\tilde{\Omega}_j} - \mathcal{I})(\mathbf{W}_{j-1})\right\|_2 \\
&\leq \sum_{j=1}^{p}\sqrt{\frac{8mn^2\beta\log(m+n)}{3q}}\|\mathbf{W}_{j-1}\|_\infty \\
&= 2\sum_{j=1}^{p}2^{-j}\sqrt{\frac{8mn^2\beta\log(m+n)}{3q}}\|\mathbf{U}_W\mathbf{V}_W^\top\|_\infty < \sqrt{\frac{32\mu rn\beta\log(m+n)}{3q}} < 1/2
\end{aligned}$$

by our assumption on q. The first line applies the triangle inequality; the second holds since $\mathbf{W}_{j-1} \in T$ for each j; the third follows because $\mathcal{P}_{T^\perp}$ is an orthogonal projection; and the final line exploits (μ, r)-coherence.

We conclude by bounding the probability of any assumed event failing. Lemma 30 implies that Eq. (9) fails to hold with probability at most $2n^{2-2\beta}$. For each k, Eq. (12) fails to hold with probability at most $2n^{2-2\beta}$ by Lemma 30, Eq. (13) fails to hold with probability at most $2n^{2-2\beta}$ by Lemma 32, and Eq. (14) fails to hold with probability at most $(m+n)^{1-2\beta}$ by Lemma 31. Hence, by the union bound, the conclusion of Theorem 29 holds with probability at least

$$1 - 2n^{2-2\beta} - \frac{3}{4}\log(n/2)(4n^{2-2\beta} + (m+n)^{1-2\beta}) \geq 1 - \frac{15}{4}\log(n)n^{2-2\beta} \geq 1 - 4\log(n)n^{2-2\beta}.$$

Appendix K. Proof of Lemma 17: Conservation of Non-Spikiness

By assumption,

$$\mathbf{L}_C\mathbf{L}_C^\top = \sum_{a=1}^{l} \mathbf{L}^{(j_a)}(\mathbf{L}^{(j_a)})^\top$$

where $\{j_1, \ldots, j_l\}$ are random indices drawn uniformly and without replacement from $\{1, \ldots, n\}$. Hence, we have that

$$\begin{aligned}\mathrm{E}\Big[\|\mathbf{L}_C\|_F^2\Big] = \mathrm{E}\Big[\mathrm{Tr}\Big[\mathbf{L}_C\mathbf{L}_C^\top\Big]\Big] &= \mathrm{Tr}\left[\mathrm{E}\left[\sum_{a=1}^{l}\mathbf{L}^{(j_a)}(\mathbf{L}^{(j_a)})^\top\right]\right] \\ &= \mathrm{Tr}\left[\sum_{a=1}^{l}\frac{1}{n}\sum_{j=1}^{n}\mathbf{L}^{(j)}(\mathbf{L}^{(j)})^\top\right] = \frac{l}{n}\mathrm{Tr}\Big[\mathbf{L}\mathbf{L}^\top\Big] = \frac{l}{n}\|\mathbf{L}\|_F^2.\end{aligned}$$

Since $\|\mathbf{L}^{(j)}\|^4 \leq m^2\|\mathbf{L}\|_\infty^4$ for all $j \in \{1, \ldots, n\}$, Hoeffding's inequality for sampling without replacement (Hoeffding, 1963, Section 6) implies

$$\begin{aligned}\mathbf{P}\Big((1-\epsilon)(l/n)\|\mathbf{L}\|_F^2 \geq \|\mathbf{L}_C\|_F^2\Big) &\leq \exp\Big(-2\epsilon^2\|\mathbf{L}\|_F^4 l^2/(n^2 l m^2\|\mathbf{L}\|_\infty^4)\Big) \\ &= \exp\big(-2\epsilon^2 l/\alpha^4(\mathbf{L})\big) \leq \delta,\end{aligned}$$

by our choice of l. Hence,

$$\sqrt{l}\frac{1}{\|\mathbf{L}_C\|_F} \leq \frac{\sqrt{n}}{\sqrt{1-\epsilon}}\frac{1}{\|\mathbf{L}\|_F}$$

with probability at least $1-\delta$. Since, $\|\mathbf{L}_C\|_\infty \leq \|\mathbf{L}\|_\infty$ almost surely, we have that

$$\alpha(\mathbf{L}_C) = \frac{\sqrt{ml}\|\mathbf{L}_C\|_\infty}{\|\mathbf{L}_C\|_F} \leq \frac{\sqrt{mn}\|\mathbf{L}\|_\infty}{\sqrt{1-\epsilon}\|\mathbf{L}\|_F} = \frac{\alpha(\mathbf{L})}{\sqrt{1-\epsilon}}$$

with probability at least $1-\delta$ as desired.

Appendix L. Proof of Theorem 18: Column Projection under Non-Spikiness

We now give a proof of Theorem 18. While the results of this section are stated in terms of i.i.d. with-replacement sampling of columns and rows, a simple argument due to (Hoeffding,

1963, Section 6) implies the same conclusions when columns and rows are sampled without replacement.

Our proof builds upon two key results from the randomized matrix approximation literature. The first relates column projection to randomized matrix multiplication:

Theorem 34 (Theorem 2 of Drineas et al. 2006b) *Let $\mathbf{G} \in \mathbb{R}^{m\times l}$ be a matrix of l columns of $\mathbf{A} \in \mathbb{R}^{m\times n}$, and let r be a nonnegative integer. Then,*

$$\|\mathbf{A} - \mathbf{G}_r\mathbf{G}_r^+\mathbf{A}\|_F \leq \|\mathbf{A} - \mathbf{A}_r\|_F + \sqrt{r}\|\mathbf{A}\mathbf{A}^\top - (n/l)\mathbf{G}\mathbf{G}^\top\|_F.$$

The second allows us to bound $\|\mathbf{A}\mathbf{A}^\top - (n/l)\mathbf{G}\mathbf{G}^\top\|_F$ in probability when entries are bounded:

Lemma 35 (Lemma 2 of Drineas et al. 2006a) *Given a failure probability $\delta \in (0, 1]$ and matrices $\mathbf{A} \in \mathbb{R}^{m\times k}$ and $\mathbf{B} \in \mathbb{R}^{k\times n}$ with $\|\mathbf{A}\|_\infty \leq b$ and $\|\mathbf{B}\|_\infty \leq b$, suppose that $\mathbf{G}$ is a matrix of l columns drawn uniformly with replacement from $\mathbf{A}$ and that $\mathbf{H}$ is a matrix of the corresponding l rows of $\mathbf{B}$. Then, with probability at least $1 - \delta$,*

$$|(\mathbf{A}\mathbf{B})_{ij} - (n/l)(\mathbf{G}\mathbf{H})_{ij}| \leq \frac{kb^2}{\sqrt{l}}\sqrt{8\log(2mn/\delta)} \quad \forall i, j.$$

Under our assumption, $\|\mathbf{M}\|_\infty$ is bounded by $\alpha/\sqrt{mn}$. Hence, Lemma 35 with $\mathbf{A} = \mathbf{M}$ and $\mathbf{B} = \mathbf{M}^\top$ guarantees

$$\|\mathbf{M}\mathbf{M}^\top - (n/l)\mathbf{C}\mathbf{C}^\top\|_F^2 \leq \frac{m^2n^2\alpha^4 8\log(2mn/\delta)}{m^2n^2l} \leq \epsilon^2/r$$

with probability at least $1 - \delta$, by our choice of l.

Now, Theorem 34 implies that

$$\begin{aligned}\|\mathbf{M} - \mathbf{C}\mathbf{C}^+\mathbf{M}\|_F \leq \|\mathbf{M} - \mathbf{C}_r\mathbf{C}_r^+\mathbf{M}\|_F &\leq \|\mathbf{M} - \mathbf{M}_r\|_F + \sqrt{r}\|\mathbf{M}\mathbf{M}^\top - (n/l)\mathbf{C}\mathbf{C}^\top\|_F \\ &\leq \|\mathbf{M} - \mathbf{L}\|_F + \epsilon\end{aligned}$$

with probability at least $1 - \delta$, as desired.

Appendix M. Proof of Theorem 20: Spikiness Master Theorem

Define $A(\mathbf{X})$ as the event that a matrix $\mathbf{X}$ is $(\alpha\sqrt{1 + \epsilon/(4\sqrt{r})})$-spiky. Since $\sqrt{1 + \epsilon/(4\sqrt{r})} \leq \sqrt{1.25}$ for all $\epsilon \in (0, 1]$ and $r \geq 1$, $\mathbf{X}$ is $(\sqrt{1.25}\alpha)$-spiky whenever $A(\mathbf{X})$ holds.

Let $\mathbf{L}_0 = [\mathbf{C}_{0,1}, \ldots, \mathbf{C}_{0,t}]$ and $\tilde{\mathbf{L}} = [\hat{\mathbf{C}}_1, \ldots, \hat{\mathbf{C}}_t]$, and define H as the event $\|\tilde{\mathbf{L}} - \hat{\mathbf{L}}^{proj}\|_F \leq \|\mathbf{L}_0 - \tilde{\mathbf{L}}\|_F + \epsilon$. When H holds, we have that

$$\begin{aligned}\|\mathbf{L}_0 - \hat{\mathbf{L}}^{proj}\|_F \leq \|\mathbf{L}_0 - \tilde{\mathbf{L}}\|_F + \|\tilde{\mathbf{L}} - \hat{\mathbf{L}}^{proj}\|_F &\leq 2\|\mathbf{L}_0 - \tilde{\mathbf{L}}\|_F + \epsilon \\ &= 2\sqrt{\textstyle\sum_{i=1}^t \|\mathbf{C}_{0,i} - \hat{\mathbf{C}}_i\|_F^2} + \epsilon,\end{aligned}$$

by the triangle inequality, and hence it suffices to lower bound $\mathbf{P}(H \cap \bigcap_i A(\mathbf{C}_{0,i}))$.

By assumption,

$$l \geq 13r\alpha^4\log(4mn/\delta)/\epsilon^2 \geq \alpha^4\log(2n/\delta)/(2\tilde{\epsilon}^2)$$

where $\tilde{\epsilon} \triangleq \epsilon/(5\sqrt{r})$. Hence, for each i, Lemma 17 implies that $\alpha(\mathbf{C}_{0,i}) \leq \alpha/\sqrt{1-\tilde{\epsilon}}$ with probability at least $1-\delta/(2n)$. Since

$$(1-\epsilon/(5\sqrt{r}))(1+\epsilon/(4\sqrt{r})) = 1+\epsilon(1-\epsilon/\sqrt{r})/(20\sqrt{r}) \geq 1$$

it follows that

$$\frac{1}{\sqrt{1-\tilde{\epsilon}}} = \frac{1}{\sqrt{1-\epsilon/(5\sqrt{r})}} \leq \sqrt{1+\epsilon/(4\sqrt{r})},$$

so that each event $A(\mathbf{C}_{0,i})$ also holds with probability at least $1-\delta/(2n)$.

Our assumption that $\|\hat{\mathbf{C}}_i\|_\infty \leq \sqrt{1.25}\alpha/\sqrt{mn}$ for all i implies that $\|\tilde{\mathbf{L}}\|_\infty \leq \sqrt{1.25}\alpha/\sqrt{mn}$. Our choice of l, with a factor of $\log(4mn/\delta)$, therefore implies that H holds with probability at least $1-\delta/2$ by Theorem 18. Hence, by the union bound,

$$\mathbf{P}(H \cap \textstyle\bigcap_i A(\mathbf{C}_{0,i})) \geq 1 - \mathbf{P}(H^c) - \sum_i \mathbf{P}(A(\mathbf{C}_{0,i})^c) \geq 1 - \delta/2 - t\delta/(2n) \geq 1-\delta.$$

To establish the DFC-RP bound, redefine H as the event $\|\tilde{\mathbf{L}} - \mathbf{L}^{rp}\|_F \leq (2+\epsilon)\|\mathbf{L}_0 - \tilde{\mathbf{L}}\|_F$. Since $p \geq 242\, r\log(14/\delta)/\epsilon^2$, H holds with probability at least $1-\delta/2$ by Corollary 9, and the DFC-RP bound follows as above.

Appendix N. Proof of Corollary 22: Noisy MC under Non-Spikiness

N.1 Proof of DFC-Proj Bound

We begin by proving the DFC-Proj bound. Let G be the event that

$$\|\mathbf{L}_0 - \hat{\mathbf{L}}^{proj}\|_F \leq 2\sqrt{c_1 \max((l/n)\nu^2, 1)/\beta} + \epsilon,$$

H be the event that

$$\|\mathbf{L}_0 - \hat{\mathbf{L}}^{proj}\|_F \leq 2\sqrt{\textstyle\sum_{i=1}^t \|\mathbf{C}_{0,i} - \hat{\mathbf{C}}_i\|_F^2} + \epsilon,$$

$A(\mathbf{X})$ be the event that a matrix $\mathbf{X}$ is $(\sqrt{1.25}\alpha)$-spiky, and, for each $i \in \{1,\ldots,t\}$, B_i be the event that $\|\mathbf{C}_{0,i} - \hat{\mathbf{C}}_i\|_F^2 > (l/n)c_1 \max\big((l/n)\nu^2, 1\big)/\beta$.

By definition, $\|\hat{\mathbf{C}}_i\|_\infty \leq \sqrt{1.25}\alpha/\sqrt{ml}$ for all i. Furthermore, we have assumed that

$$\begin{aligned} l &\geq 13(c_3+1)\sqrt{\frac{(m+n)\log(m+n)\beta}{s}} nr\alpha^4 \log(4mn)/\epsilon^2 \\ &\geq 13r\alpha^4(\log(4mn) + c_3\log(m+n))/\epsilon^2 \geq 13r\alpha^4 \log(4mn(m+l)^{c_3})/\epsilon^2. \end{aligned}$$

Hence the Spikiness Master Theorem (Theorem 20) guarantees that, with probability at least $1-\exp(-c_3\log(m+l))$, H holds and the event $A(\mathbf{C}_{0,i})$ holds for each i. Since G holds whenever H holds and B_i^c holds for each i, we have

$$\begin{aligned} \mathbf{P}(G) &\geq \mathbf{P}(H \cap \textstyle\bigcap_i B_i^c) \geq \mathbf{P}(H \cap \bigcap_i A(\mathbf{C}_{0,i}) \cap \bigcap_i B_i^c) \\ &= \mathbf{P}(H \cap \textstyle\bigcap_i A(\mathbf{C}_{0,i}))\mathbf{P}(\bigcap_i B_i^c \mid H \cap \bigcap_i A(\mathbf{C}_{0,i})) \\ &= \mathbf{P}(H \cap \textstyle\bigcap_i A(\mathbf{C}_{0,i}))(1 - \mathbf{P}(\bigcup_i B_i \mid H \cap \bigcap_i A(\mathbf{C}_{0,i}))) \\ &\geq (1-\exp(-c_3\log(m+l)))(1 - \textstyle\sum_i \mathbf{P}(B_i \mid A(\mathbf{C}_{0,i}))) \\ &\geq 1 - (c_2+1)\exp(-c_3\log(m+l)) - \textstyle\sum_i \mathbf{P}(B_i \mid A(\mathbf{C}_{0,i})). \end{aligned}$$

To prove our desired claim, it therefore suffices to show

$$\mathbf{P}(B_i \mid A(\mathbf{C}_{0,i})) \leq (c_2 + 1)\exp(-c_3 \log(m + l))$$

for each i.

For each i, let D_i be the event that $s_i < 1.25\alpha^2\beta(n/l)r(m + l)\log(m + l)$, where s_i is the number of revealed entries in $\mathbf{C}_{0,i}$. Since $\text{rank}(\mathbf{C}_{0,i}) \leq \text{rank}(\mathbf{L}_0) = r$ and $\|\mathbf{C}_{0,i}\|_F \leq \|\mathbf{L}_0\|_F \leq 1$, Corollary 19 implies that

$$\begin{aligned}\mathbf{P}(B_i \mid A(\mathbf{C}_{0,i})) &\leq \mathbf{P}(B_i \mid A(\mathbf{C}_{0,i}), D_i^c) + \mathbf{P}(D_i \mid A(\mathbf{C}_{0,i})) \\ &\leq c_2 \exp(-c_3 \log(m + l)) + \mathbf{P}(D_i). \end{aligned} \tag{15}$$

Further, since the support of $\mathbf{S}_0$ is uniformly distributed and of cardinality s, the variable s_i has a hypergeometric distribution with $\mathbf{E}(s_i) = \frac{sl}{n}$ and hence satisfies Hoeffding's inequality for the hypergeometric distribution (Hoeffding, 1963, Section 6):

$$\mathbf{P}(s_i \leq \mathbf{E}(s_i) - st) \leq \exp\bigl(-2st^2\bigr).$$

Our assumption on l implies that

$$\begin{aligned}\frac{l}{n} &\geq 169(c_3 + 1)^2\alpha^8\beta\frac{n}{ls}r^2(m + n)\log(m + n)\log^2(4mn)/\epsilon^4 \\ &\geq 1.25\alpha^2\beta\frac{n}{ls}r(m + l)\log(m + l) + \sqrt{c_3 \log(m + l)/(2s)},\end{aligned}$$

and therefore

$$\begin{aligned}\mathbf{P}(D_i) &= \mathbf{P}\left(s_i < \mathbf{E}(s_i) - s\left(\frac{l}{n} - 1.25\alpha^2\beta\frac{n}{ls}r(m + l)\log(m + l)\right)\right) \\ &= \mathbf{P}\left(s_i < \mathbf{E}(s_i) - s\sqrt{c_3 \log(m + l)/(2s)}\right) \\ &\leq \exp(-2sc_3 \log(m + l)/(2s)) = \exp(-c_3 \log(m + l)).\end{aligned}$$

Combined with Eq. (15), this yields $\mathbf{P}(B_i \mid A(\mathbf{C}_{0,i})) \leq (c_2 + 1)\exp(-c_3 \log(m + l))$ for each i, and the DFC-Proj result follows.

N.2 Proof of DFC-RP Bound

Let G be the event that

$$\|\mathbf{L}_0 - \hat{\mathbf{L}}^{rp}\|_F \leq (2 + \epsilon)\sqrt{c_1 \max((l/n)\nu^2, 1)/\beta}$$

and H be the event that

$$\|\mathbf{L}_0 - \hat{\mathbf{L}}^{rp}\|_F \leq (2 + \epsilon)\sqrt{\textstyle\sum_{i=1}^t \|\mathbf{C}_{0,i} - \hat{\mathbf{C}}_i\|_F^2}.$$

Since $p \geq 242\ r\log(14(m + l)^{c_3})/\epsilon^2$, the DFC-RP bound follows in an identical manner from the Spikiness Master Theorem (Theorem 20).

References

A. Agarwal, S. Negahban, and M. J. Wainwright. Noisy matrix decomposition via convex relaxation: Optimal rates in high dimensions. In *International Conference on Machine Learning*, 2011.

E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM*, 58(3):1–37, 2011.

E.J. Candès and Y. Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6): 925 –936, 2010.

V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky. Sparse and low-rank matrix decompositions. In *Allerton Conference on Communication, Control, and Computing*, 2009.

Y. Chen, H. Xu, C. Caramanis, and S. Sanghavi. Robust matrix completion and corrupted columns. In *International Conference on Machine Learning*, 2011.

A. Das, M. Datar, A. Garg, and S. Rajaram. Google news personalization: scalable online collaborative filtering. In *WWW*, 2007.

P. Drineas, R. Kannan, and M. W. Mahoney. Fast monte carlo algorithms for matrices ii: Computing a low-rank approximation to a matrix. *SIAM J. Comput.*, 36(1):158–183, 2006a.

P. Drineas, R. Kannan, and M. W. Mahoney. Fast monte carlo algorithms for matrices i: Approximating matrix multiplication. *SIAM J. Comput.*, 36(1):132–157, 2006b.

P. Drineas, M. W. Mahoney, and S. Muthukrishnan. Relative-error CUR matrix decompositions. *SIAM Journal on Matrix Analysis and Applications*, 30:844–881, 2008.

B. Recht F. Niu, C. Ré, and S. J. Wright. Hogwild!: A lock-free approach to parallelizing stochastic gradient descent. In *NIPS*, 2011.

A. Frieze, R. Kannan, and S. Vempala. Fast Monte-Carlo algorithms for finding low-rank approximations. In *Foundations of Computer Science*, 1998.

R. Gemulla, E. Nijkamp, P. J. Haas, and Y. Sismanis. Large-scale matrix factorization with distributed stochastic gradient descent. In *KDD*, 2011.

S. A. Goreinov, E. E. Tyrtyshnikov, and N. L. Zamarashkin. A theory of pseudoskeleton approximations. *Linear Algebra and its Applications*, 261(1-3):1 – 21, 1997.

D. Gross and V. Nesme. Note on sampling without replacing from a finite collection of matrices. *CoRR*, abs/1001.2738, 2010.

N. Halko, P. G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011.

W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

P. D. Hoff. Bilinear mixed-effects models for dyadic data. *Journal of the American Statistical Association*, 100:286–295, March 2005.

D. Hsu. `http://www.cs.columbia.edu/~djhsu/papers/randmatrix-errata.txt`, 2012.

D. Hsu, S. Kakade, and T. Zhang. Tail inequalities for sums of random matrices that depend on the intrinsic dimension. *Electron. Commun. Probab.*, 17:no. 14, 1–13, 2012.

W. B. Johnson and J. Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary Mathematics*, 26:189–206, 1984.

R. H. Keshavan, A. Montanari, and S. Oh. Matrix completion from noisy entries. *Journal of Machine Learning Research*, 99:2057–2078, 2010.

Y. Koren, R. M. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8):30–37, 2009.

S. Kumar, M. Mohri, and A. Talwalkar. On sampling-based approximate spectral decomposition. In *International Conference on Machine Learning*, 2009a.

S. Kumar, M. Mohri, and A. Talwalkar. Ensemble Nyström method. In *Advances in Neural Information Processing Systems*, 2009b.

E. Liberty. *Accelerated Dense Random Projections*. Ph.D. thesis, computer science department, Yale University, New Haven, CT, 2009.

Z. Lin, M. Chen, L. Wu, and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. UIUC Technical Report UILU-ENG-09-2215, 2009a.

Z. Lin, A. Ganesh, J. Wright, L. Wu, M. Chen, and Y. Ma. Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix. UIUC Technical Report UILU-ENG-09-2214, 2009b.

S. Ma, D. Goldfarb, and L. Chen. Fixed point and bregman iterative methods for matrix rank minimization. *Mathematical Programming*, 128(1-2):321–353, 2011.

L. Mackey, A. Talwalkar, and M. I. Jordan. Divide-and-conquer matrix factorization. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. C. N. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 1134–1142. 2011.

M. W. Mahoney and P. Drineas. Cur matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences*, 106(3):697–702, 2009.

K. Min, Z. Zhang, J. Wright, and Y. Ma. Decomposing background topics from keywords by principal component pursuit. In *Conference on Information and Knowledge Management*, 2010.

M. Mohri and A. Talwalkar. Can matrix coherence be efficiently and accurately estimated? In *Conference on Artificial Intelligence and Statistics*, 2011.

Y. Mu, J. Dong, X. Yuan, and S. Yan. Accelerated low-rank visual recovery by random projection. In *Conference on Computer Vision and Pattern Recognition*, 2011.

S. Negahban and M. J. Wainwright. Restricted strong convexity and weighted matrix completion: Optimal bounds with noise. *J. Mach. Learn. Res.*, 13:1665–1697, 2012.

E. J. Nyström. Über die praktische auflösung von integralgleichungen mit anwendungen auf randwertaufgaben. *Acta Mathematica*, 54(1):185–204, 1930.

C. H. Papadimitriou, H. Tamaki, P. Raghavan, and S. Vempala. Latent Semantic Indexing: a probabilistic analysis. In *Principles of Database Systems*, 1998.

Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma. Rasl: Robust alignment by sparse and low-rank decomposition for linearly correlated images. In *Conference on Computer Vision and Pattern Recognition*, 2010.

B. Recht. Simpler approach to matrix completion. *J. Mach. Learn. Res.*, 12:3413–3430, 2011.

B. Recht and C. Ré. Parallel stochastic gradient algorithms for large-scale matrix completion. In *Optimization Online*, 2011.

V. Rokhlin, A. Szlam, and M. Tygert. A randomized algorithm for Principal Component Analysis. *SIAM Journal on Matrix Analysis and Applications*, 31(3):1100–1124, 2009.

A. Talwalkar and A. Rostamizadeh. Matrix coherence and the Nyström method. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, 2010.

K. Toh and S. Yun. An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems. *Pacific Journal of Optimization*, 6(3):615–640, 2010.

M. Tygert. http://www.mathworks.com/matlabcentral/fileexchange/21524-principal-component-analysis, 2009.

J. Wang, Y. Dong, X. Tong, Z. Lin, and B. Guo. Kernel Nyström method for light transport. *ACM Transactions on Graphics*, 28(3), 2009.

C.K. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems*, 2000.

H.-F. Yu, C.-J. Hsieh, S. Si, and I. Dhillon. Scalable coordinate descent approaches to parallel matrix factorization for recommender systems. In *ICDM*, 2012.

Y. Zhou, D. Wilkinson, R. Schreiber, and R. Pan. Large-scale parallel collaborative filtering for the netflix prize. In *Proceedings of the 4th international conference on Algorithmic Aspects in Information and Management*, 2008.

Z. Zhou, X. Li, J. Wright, E. J. Candès, and Y. Ma. Stable principal component pursuit. In *IEEE International Symposium on Information Theory Proceedings (ISIT)*, pages 1518 –1522, 2010.

Journal of Machine Learning Research 16 (2015) 961-992 Submitted 10/13; Revised 11/14; Published 5/15

Combined ℓ_1 and Greedy ℓ_0 Penalized Least Squares for Linear Model Selection*

Piotr Pokarowski POKAR@MIMUW.EDU.PL
Faculty of Mathematics, Informatics and Mechanics
University of Warsaw
Banacha 2, 02-097 Warsaw, Poland

Jan Mielniczuk MIEL@IPIPAN.WAW.PL
Faculty of Mathematics and Information Science
Warsaw University of Technology
Koszykowa 75, 00-662 Warsaw, Poland

Institute of Computer Science
Polish Academy of Sciences
Jana Kazimierza 5, 01-248 Warsaw, Poland

Editor: Tong Zhang

Abstract

We introduce a computationally effective algorithm for a linear model selection consisting of three steps: screening–ordering–selection (SOS). Screening of predictors is based on the thresholded Lasso that is ℓ_1 penalized least squares. The screened predictors are then fitted using least squares (LS) and ordered with respect to their $|t|$ statistics. Finally, a model is selected using greedy generalized information criterion (GIC) that is ℓ_0 penalized LS in a nested family induced by the ordering. We give non-asymptotic upper bounds on error probability of each step of the SOS algorithm in terms of both penalties. Then we obtain selection consistency for different (n, p) scenarios under conditions which are needed for screening consistency of the Lasso. Our error bounds and numerical experiments show that SOS is worth considering alternative for multi-stage convex relaxation, the latest quasiconvex penalized LS. For the traditional setting $(n > p)$ we give Sanov-type bounds on the error probabilities of the ordering–selection algorithm. It is surprising consequence of our bounds that the selection error of greedy GIC is asymptotically not larger than of exhaustive GIC.

Keywords: linear model selection, penalized least squares, Lasso, generalized information criterion, greedy search, multi-stage convex relaxation

1. Introduction

Literature concerning linear model selection has been lately dominated by analysis of the *least absolute shrinkage and selection operator* (Lasso) that is ℓ_1 penalized least squares for the 'large p - small n scenario', where n is number of observations and p is number of all predictors. For a broad overview of the subject we refer to Bühlmann and van de Geer (2011). It is known that consistency of selection based on the Lasso requires strong regularity

*. This paper is dedicated to Jacek Koronacki on the occasion of his 70th birthday.

of an experimental matrix named *irrepresentable conditions* which are rather unlikely to hold in practice (Meinshausen and Bühlmann, 2006; Zhao and Yu, 2006). However, consistency of the Lasso predictors or consistency of the Lasso estimators of the linear model parameters is proved under weaker assumptions such as the *restricted isometry property* (RIP). The last condition means that singular values of normalized experimental submatrices corresponding to small sets of predictors are uniformly bounded away from zero and infinity. Under those more realistic conditions and provided that a certain lower bound on the absolute values of model parameters called *beta-min condition* holds, the Lasso leads to consistent screening, that is the set of nonzero Lasso coefficients S contains with large predetermined probability the uniquely defined true model T. This property explains Bühlmann's suggestion that one should interpret the second 's' in 'Lasso' as 'screening' rather than 'selection' (see discussion of Tibshirani, 2011) and the task is now to remove the spurious selected predictors. To this aim two-stage procedures as the adaptive or the thresholded Lasso have been proposed (cf. Zou, 2006; Huang et al., 2008; Meinshausen and Yu, 2009; Zhou, 2009, 2010; van de Geer et al., 2011). They yield selection consistency under strong version of the beta-min condition and without such strengthening tend to diminish the number of selected spurious predictors, but, similarly to the Lasso they yield screening consistency only. Alternative approaches require minimization of *least squares* (LS) penalized by quasiconvex functions that are closer to the ℓ_0 penalty then ℓ_1 (Fan and Li, 2001; Zou and Li, 2008; Zhang, 2010a,b; Zhang and Zhang, 2012; Huang and Zhang, 2012; Zhang, 2013; Wang et al., 2014). These methods lead to consistent selection under RIP and considerably weaker version of the beta-min condition, nevertheless are more computationally demanding.

Regularization is required when a model matrix is not a full rank or when $n < p$, but for the traditional regression when an experimental plan is of full rank and $n > p$ it is possible to construct a computationally effective and selection consistent two-stage *ordering–selection* (OS) procedure, as follows. First, a full model F using LS is fitted, predictors are ordered with respect to their $|t|$ statistics from the fit and finally, a submodel of F in a nested family pertaining to the ordering is selected using thresholding as in Rao and Wu (1989), Bunea et al. (2006) or *generalized information criterion* (GIC) as in Zheng and Loh (1995). The OS algorithm can be treated as *greedy* ℓ_0 penalized LS because it requires computing a criterion function for $2p$ models only instead of all 2^p models. Frequently, sufficient conditions on an experimental plan and a vector of true coefficients for consistency of such procedures are stated in terms of the *Kullback-Leibler divergence* (KL) of the true model from models which lack at least one true predictor (Zheng and Loh, 1995; Shao, 1998; Chen and Chen, 2008; Casella et al., 2009; Pötscher and Schneider, 2011; Luo and Chen, 2013). In particular, a bound on the probability of selection error in Shao (1998) closely resembles the Sanov theorem in information theory on bounds of probability of a non-typical event using the KL divergence.

In our contribution we introduce a computationally effective three-step algorithm for linear model selection based on a *screening–ordering–selection* (SOS) scheme. Screening of predictors is based on a version of the thresholded Lasso proposed by Zhou (2009, 2010) and yields the screening set S such that $|S| \leq n$. Next, an implementation of the OS algorithm described above proposed by Zheng and Loh (1995) is applied. We give non-asymptotic upper bounds on error probability of each step of the SOS algorithm in terms of the Lasso and GIC penalties (Theorem 1). As a consequence of proved bounds we obtain

selection consistency for different (n, p) scenarios under weak conditions which are sufficient for screening consistency of the Lasso. Our assumptions allow for strong correlation between predictors, in particular replication of spurious predictors is possible.

The SOS algorithm is an improvement of the new version of the thresholded Lasso and turns out to be a promising competitor to *multi-stage convex relaxation* (MCR), the latest quasiconvex penalized LS (Zhang, 2010b, 2013). The condition on correlation of predictors assumed there seems to be stronger than ours, whereas the beta-min condition may be weaker (Section 5). In our simulations for $|T| \ll n \ll p$ scenario, SOS was faster and more accurate than MCR (Section 8).

For case $n > p$ we also give a bound on probability of selection error of the OS algorithm. Our bound in this case is more general than in Shao (1998) as we allow ordering of predictors, $p = p_n \to \infty$, $|T| = |T_n| \to \infty$ or the GIC penalty may be of order n (Theorem 2). It is surprising consequence of Theorems 1-2 that the probability of selection error of greedy GIC is asymptotically not larger than of exhaustive GIC. Thus employment of greedy search dramatically decreases computational cost of l_0 penalized LS minimization without increasing selection error probability.

As a by-product we obtained a strengthened version of the nonparametric sparse oracle inequality for the Lasso proved by Bickel et al. (2009) and, as its consequence, more tight bounds on prediction and estimation error (Theorem 4). We simplified and strengthened an analogous bound for the thresholded Lasso given by Zhou (2009, 2010) (Theorem 1 part T1). It is worth noticing that all results are proved simultaneously for two versions of the algorithm: for the Lasso used in practice when a response is centered and predictors are standardized as well as for its formal version for which an intercept corresponds to a dummy predictor.

The paper is organized as follows. In Section 2 the SOS algorithm is introduced and in Section 3 we study properties of geometric characteristics pertaining to an experimental matrix and a vector of coefficients which are related to identifiability of a true model. Section 4 contains our main results that is bounds on selection error probabilities for the SOS and OS algorithm. In Section 5 we briefly discuss the MCR algorithm and compare error bounds for SOS and MCR. Section 6 treats properties of post-model selection estimators pertaining to SOS and MCR. Section 7 contains improved bounds on the Lasso estimation and prediction. Section 8 presents a simulational study. Concluding remarks are given in Section 9. Appendix contains detailed proofs of the stated results.

2. Selection Algorithm

The aim of this section is to describe the proposed selection algorithm. As in the first step of the algorithm we use the Lasso estimator to screen predictors and since in the literature there exist two versions of the Lasso for the linear model which differ in the treatment of the intercept, we start this section by defining two parametrizations of the linear model related to these versions of the Lasso. Next we state a general definition encompassing both cases, present our implementation of the SOS scheme and finally we discuss its computational complexity.

2.1 Linear Regression Model Parametrizations

We consider a general regression model of real-valued responses having the following structure

$$y_i = \mu(x_{i.}) + \varepsilon_i, \qquad i = 1, 2, \ldots, n,$$

where $\varepsilon_1, \ldots, \varepsilon_n$ are iid $N(0, \sigma^2)$, $x_{i.} \in \mathbf{R}^p$, and $p = p_n$ may depend on n. In a vector form we have

$$y = \mu + \varepsilon, \tag{1}$$

where $\mu = (\mu(x_{1.}), \ldots, \mu(x_{n.}))^T, \varepsilon = (\varepsilon_1, \ldots, \varepsilon_n)^T$ and $y = (y_1, \ldots, y_n)^T$.

Let $X = [x_{1.}, \ldots, x_{n.}]^T = [x_1, \ldots, x_p]$ be the $n \times p$ matrix of experiment. We consider two linear parametrizations of (1). The first parametrization is:

$$\mu = \alpha^* + X\beta^*, \tag{2}$$

where $\alpha^* \in \mathbf{R}$ is an intercept and $\beta^* \in \mathbf{R}^p$ is a vector of coefficients corresponding to predictors. The second parametrization is

$$\mu = X\beta^*, \tag{3}$$

where the intercept is either set to 0 or is incorporated into vector β^* and treated in the same way as all other coefficients in the linear model. In order to treat both parametrizations in the same way we write $\mu = \tilde{X}\tilde{\beta}^*$ where, with $\mathbb{1}_n$ denoting a column of ones, $\tilde{X} = [\mathbb{1}_n, X]$ and $\tilde{\beta}^* = (\alpha^*, \beta^{*T})^T$ in the case of (2) and $\tilde{X} = X$ and $\tilde{\beta}^* = \beta^*$ in the case of (3). We note that (3) is convenient for theoretical considerations and simulations on synthetic data, but (2) is natural for real data applications and occurs as a default option in popular statistical software.

Let $J \subseteq \{1, 2, \ldots, p\} = F$ be an arbitrary subset of the full model F and $|J|$ the number of its elements, X_J is a submatrix of X with columns having indices in J, β_J is a subvector of β with columns having indices in J. Moreover, let $\tilde{X}_J = [\mathbb{1}_n, X_J]$ and $\tilde{\beta}_J = (\alpha, \beta_J^T)^T$ in the case of (2) or $\tilde{X}_J = X_J$ and $\tilde{\beta}_J = \beta_J$ in the case of (3). $\tilde{H}_J$ will stand for a projection matrix onto the subspace spanned by columns of $\tilde{X}_J$. Linear model pertaining to predictors being columns of X_J will be frequently identified as J. We will also denote by $T = T_n$ a true model that is a model such that $T = \mathrm{supp}(\beta^*) = \{j \in F : \beta_j^* \neq 0\}$ for some β^* such that $\mu = \tilde{X}\tilde{\beta}^*$. The uniqueness of T and β^* for a given n will be discussed in Section 3.

2.2 Practical and Formal Lasso

The Lasso introduced in Tibshirani (1996) is a popular method of estimating β^* in the linear model. For discussion of properties of the Lasso see for example Tibshirani (2011) and Bühlmann and van de Geer (2011). When using the Lasso for data analytic purposes parametrization (2) is considered, vector of responses y is centered and columns of X are standardized. The standardization step is usually omitted in formal analysis in which parametrization (3) is assumed, α is taken to be 0 and X consists of meaningful predictors only, without column of ones corresponding to intercept. Alternatively, columns of X are normalized by their norms (see for example formula 2.1 in Bickel et al., 2009). Here, in order to accommodate considered approaches in one definition we introduce a general form

of the Lasso. Let H_0 be an $n \times n$ projection matrix, where H_0 is specified as a vector centering matrix $\mathbb{I}_n - \mathbb{1}_n \mathbb{1}_n^T/n$ in the case of the applied version of the Lasso pertaining to parametrization (2) and the identity matrix $\mathbb{I}_n$ for the formal Lasso corresponding to (3). Moreover, let

$$D = \mathrm{diag}(||H_0 x_j||)_{j=1}^p, \quad X_0 = H_0 X D^{-1}, \quad X_0 = [x_{01}, \ldots, x_{0p}], \quad y_0 = H_0 y \tag{4}$$

and $\theta^* = D\beta^*$, $\mu_0 = H_0\mu$. For estimation of β^*, data (X_0, y_0) will be used. Note that for the first choice of orthogonal projection in the definition of X_0 columns in X are normalized by their norms whereas for the second they are standardized (centered and divided by their standard deviations). Consider the case of (2) and denote by H_{0J} projection onto $\mathrm{sp}\{(H_0 x_j)_{j\in J}\}$. Observe that as $\mathrm{sp}\{\mathbb{1}_n, (x_j)_{j\in J}\} = \mathrm{sp}\{\mathbb{1}_n\} \oplus \mathrm{sp}\{(H_0 x_j)_{j\in J}\}$ and consequently $\tilde{H}_J = H_{0J} + \mathbb{1}_n \mathbb{1}_n^T/n$, we have that

$$\mathbb{I}_n - \tilde{H}_J = (\mathbb{I}_n - H_{0J}) H_0. \tag{5}$$

The above equality trivially holds also in the case of (3).

For $a = (a_j) \in \mathbf{R}^k$, let $|a| = \sum_{j=1}^k |a_j|$ and $||a|| = (\sum_{j=1}^k a_j^2)^{1/2}$ be ℓ_1 and ℓ_2 norms, respectively. As J may be viewed as sequence of zeros and ones on F, $|J|$ denotes cardinality of J.

General form of the Lasso estimator of β is defined as follows

$$\hat{\beta} = \mathrm{argmin}_\beta\{||H_0(y - X\beta)||^2 + 2r_L|D\beta|\} = D^{-1}(\mathrm{argmin}_\theta\{||y_0 - X_0\theta||^2 + 2r_L|\theta|\}), \tag{6}$$

where a parameter $r_L = r_{nL}$ is a penalty on l_1 norm of a potential estimator of β. Thus in the case of parametrization (2) the Lasso estimator of β may be defined without using extended matrix $\tilde{X}$ by applying H_0 to $y - X\beta$ that is by centering it. In the case of parametrization (3) $H_0 = \mathbb{I}_n$ and the usual definition of the Lasso used in formal analysis is obtained. We remark that the approaches used in theoretical considerations for which columns of X are not normalized as in Bühlmann and van de Geer (2011) or Zhang (2013) formally correspond to (6) with $H_0 = \mathbb{I}_n$ and $D = dI_p$, where $d = \max_{1\le j\le p} ||x_j||$.

Note that in the case of parametrization (2) $\hat{\beta}$ is subvector corresponding to β of the following minimizer

$$\mathrm{argmin}_{\tilde{\beta}}\{||y - \tilde{X}\tilde{\beta}||^2 + 2r_L|D\beta|\} = \mathrm{argmin}_{\alpha,\beta}\{||y - \alpha\mathbb{1}_n - X\beta||^2 + 2r_L|D\beta|\}, \tag{7}$$

where the equality of minimal values of expressions appearing in (6) and (7) is obtained when the expression $||y - \alpha\mathbb{1}_n - X\beta||^2$ is minimized with respect to α for fixed β. However, omitting centering projection H_0 in (6) when the first column of X consists of ones and corresponds to intercept, leads to lack of invariance of $\hat{\beta}$ when the data are shifted by a constant and yields different estimates that those used in practice. This is a difference between the Lasso and the LS estimator: LS estimator has the same form regardless of which of the two parametrizations (2) or (3) is applied. Using (5) we have for the LS estimator $\hat{\beta}_J^{LS}$ in model J that the sum of squared residuals for the projection $\tilde{H}y$ equals

$$R_J = ||(\mathbb{I}_n - \tilde{H}_J)y||^2 = ||(\mathbb{I}_n - H_{0J})y_0||^2 = ||y_0 - X_{0J}\hat{\theta}_J^{LS}||^2 \tag{8}$$

and

$$\hat{\beta}_J^{LS} = D^{-1}\hat{\theta}_J^{LS}, \quad \hat{\theta}_J^{LS} = \mathrm{argmin}_{\theta_J}||y_0 - X_{0J}\theta_J||^2.$$

2.3 Implementation of the Screening–Ordering–Selection Scheme

The SOS algorithm which is the main subject of the paper is the following implementation of the SOS scheme.

Algorithm *(SOS)*
Input: y, X and r_L, b, r.

Screening. Compute the Lasso estimator $\hat{\beta} = D^{-1}\hat{\theta}, \quad \hat{\theta} = (\hat{\theta}_1, \dots \hat{\theta}_p)^T$ with a penalty parameter r_L and set $S_0 = \{j : |\hat{\theta}_j| > b\}, \quad B = b(|S_0| \vee 1)^{1/2}, \quad S_1 = \{j : |\hat{\theta}_j| > B\}$.

Ordering. Fit the model S_1 by ordinary LS and order predictors $\hat{O} = (j_1, j_2, \dots, j_{|S_1|})$ using values of corresponding squared t statistics $t^2_{j_1} \geq t^2_{j_2} \geq \dots \geq t^2_{j_{|S_1|}}$.

Selection. In the nested family $\mathcal{G} = \{\emptyset, \{j_1\}, \{j_1, j_2\}, \dots, S_1\}$ choose a model $\hat{T} \equiv \hat{T}_{S_1,\hat{O}}$ according to the *generalized information criterion* (GIC) $\hat{T} = \mathrm{argmin}_{J\in\mathcal{G}}\{R_J + |J|r\}$, where $r = r_n$ is a penalty pertaining to GIC.

Output: $\hat{T}_{SOS} = \hat{T}$, $\hat{\beta}^{SOS} = \hat{\beta}^{LS}_{\hat{T}}$.

The OS algorithm is intended for the case $p < n$ and is a special case of SOS for which S_1 is taken equal to F.

We note that empty set in the definition of $\mathcal{G}$ corresponds to $\mu = 0$ in the case of parametrization (3) and $\mu = \alpha^*$ in the case of (2). It is easy to check also that

$$\frac{t_j^2}{n - |S_1|} = \frac{R_{S_1\setminus\{j\}} - R_{S_1}}{R_{S_1}}, \tag{9}$$

thus ordering with respect to decreasing values of (t_j^2) in the second step of the procedure is the same as ordering of $(R_{S_1\setminus\{j\}})$ in decreasing order.

2.4 Computational Complexity of the SOS Algorithm

There are many approximate algorithms for the Lasso estimator (6) as quadratic program solvers or coordinate descent in Friedman et al. (2010). The popular LARS method proposed in Efron et al. (2004) can be used to compute exactly, in finitely many steps, the whole Lasso regularized solution path which is piecewise linear with respect to r_L. It has been shown recently in Mairal and Yu (2012) that, in the worst case, the number of linear segments of this path is exactly $(3^p+1)/2$, so the overall computational cost of the Lasso is $O(3^p pn)$, see Rosset and Zhu (2007). Hence, by the most popular criterion of computational complexity LARS does not differ from, for example, an exhaustive search for the ℓ_0 penalized LS problem. However, experience with data suggests that the number of linear segments of the LARS regularization path is typically $O(n)$, so LARS execution requires $O(np\min(n, p))$ flops, see Rosset and Zhu (2007) and Bühlmann and van de Geer (2011), chapter 2.12. Thus taking into account the result in Mairal and Yu (2012) on uniform approximation of the Lasso regularization paths, for typical data set the Lasso may be considered computationally efficient (cf. also discussion on the page 7 in Zhang (2013)).

In Section 4 we will discuss conditions on X and β^*_T, under which S_1 includes a unique true model T and $|S_1| \leq n$ or even $|S_1| \leq 4|T|$ with high probability. In this case we can use LS to fit a linear model, thus the ordering step takes $O(n|S_1|^2)$ calculations by the

QR decomposition of the matrix X_{0S_1}. Computing $(R_J)_{J\in\mathcal{G}}$ in the selection step demands also only one QR decomposition of X_{0S_1} with columns ordered according to $\hat{O}$. Indeed, let $X_{0S_1} = QW$, where an orthogonal matrix $Q = [q_1, \ldots, q_{|S_1|}]$. The following iterative procedure can be used

$$R_\emptyset = ||y_0||^2; \quad \textbf{for} \quad k = 1, \ldots, |S_1| \quad \textbf{do} \quad R_{\{1,\ldots,k\}} = R_{\{1,\ldots,k-1\}} - (q_k^T y_0)^2 \quad \textbf{endfor}.$$

Observe, that from (9) the ordering part demands GIC only for $|S_1|$ models that is for $S_1 \setminus \{j\},\ j \in S_1$. Thus two last parts of the SOS algorithm or, equivalently, the OS algorithm demands GIC only for $2|S_1|$ models instead of all $2^{|S_1|}$ and we can call it *greedy* ℓ_0 penalized LS.

We conclude that the SOS algorithm is computationally efficient and the most time expensive part of it is the screening. The same conclusion follows from our simulations described in Section 8.

3. A True Model Identifiability

In this section we consider two types of linear model characteristics which will be used to quantify the difficulty of selection or, equivalently, a true model identifiability problem, and we study the interplay between them.

3.1 Kullback-Leibler Divergences

Let T be given true model that is $T \subseteq F$ such that $\mu = \tilde{X}\tilde{\beta}^* = \tilde{X}_T\tilde{\beta}^*_T$ and $T = \mathrm{supp}(\beta^*_T) = \{j \in F : \beta^*_{j,T} \neq 0\}$. For $J \subseteq F$ define

$$\delta(T \parallel J) = ||(\mathbb{I}_n - \tilde{H}_J)\tilde{X}_T\tilde{\beta}^*_T||^2.$$

In view of (5) we obtain

$$\delta(T \parallel J) = ||(\mathbb{I}_n - H_{0J})H_0\tilde{X}_T\tilde{\beta}^*_T||^2 = ||(\mathbb{I}_n - H_{0J})H_0X_T\beta^*_T||^2 = ||(\mathbb{I}_n - H_{0J})X_{0T}\theta^*_T||^2. \quad (10)$$

Let $KL(\tilde{\beta}^*_T \parallel \tilde{\beta}_J) = \mathbf{E}_{\tilde{\beta}^*_T}\log(f_{\tilde{\beta}^*_T}/f_{\tilde{\beta}_J})$ be the *Kullback-Leibler divergence* of the normal density $f_{\tilde{\beta}^*_T}$ of $N(\tilde{X}_T\tilde{\beta}^*_T, \sigma^2\mathbb{I}_n)$ from the normal density $f_{\tilde{\beta}_J}$ of $N(\tilde{X}_J\tilde{\beta}_J, \sigma^2\mathbb{I}_n)$. Let $\Sigma = X_0^TX_0$ be a *coherence matrix* if H_0 is the identity matrix and *a correlation matrix* if $H_0 = \mathbb{I}_n - \mathbb{1}_n\mathbb{1}_n^T/n$. Let Σ_J stands for a submatrix of Σ with columns having indices in J and let $\lambda_{min}(\Sigma_J),\ \lambda_{max}(\Sigma_J)$ denote extremal eigenvalues of Σ_J. The following proposition lists the basic properties of the parameter δ. Observe also that $\delta(T \parallel J)$ is a parameter of non-centrality of χ^2 distribution of R_J that is $R_J \sim \chi^2_{n-|J|}(\delta(T \parallel J))$.

Proposition 1

$$(i) \quad \delta(T \parallel J) = 2\sigma^2 \min_{\tilde{\beta}_J} KL(\tilde{\beta}^*_T \parallel \tilde{\beta}_J) = 2\sigma^2 \min_{\tilde{\beta}_J} KL(\tilde{\beta}_J \parallel \tilde{\beta}^*_T).$$

$$(ii) \quad \delta(T \parallel J) = \min_{\theta_J} \left|\left| [X_{0,T\setminus J}, X_{0,J}] \begin{pmatrix} \theta^*_{T\setminus J} \\ \theta_J \end{pmatrix} \right|\right|^2 \geq \lambda_{min}(\Sigma_{J\cup T})||\theta^*_{T\setminus J}||^2 \quad (11)$$

The following scaled Kullback-Leibler divergence will be employed in our main results in Section 4.

$$\delta(T,s)=\min_{j\in T,J\supseteq T,|J|\leq s}\delta(T\parallel J\setminus\{j\}).$$

This coefficient was previously used to prove selection consistency in Zheng and Loh (1995); Chen and Chen (2008); Luo and Chen (2013) and to establish asymptotic law of post-selection estimators in Pötscher and Schneider (2011). Similar coefficients appear in proofs of selection consistency in Shao (1998) and Casella et al. (2009). Obviously, $\delta(T,s)$ is a nonincreasing function of s.

Identifiability of a true model is stated in the proposition below in terms of

$$\delta(T)=\min_{J\not\supseteq T,|J|\leq|T|}\delta(T\parallel J).$$

Proposition 2 *There exists at most one true model T such that $\delta(T)>0$.*

Assume by contradiction that T' is a different true model, that is we have $T'=\mathrm{supp}(\tilde{\beta})$ for some $\tilde{\beta}$ such that $\mu=\tilde{X}\tilde{\beta}$. Then by symmetry we can assume $|T|\leq|T'|$. Hence $|T'\setminus T|>0$ and $\delta(T')\leq\delta(T'\parallel T)=0$.
It is easy to see that if $\delta(T)>0$ then columns of X_T are linearly independent and, consequently, there exists at most one $\tilde{\beta}_T^*$ such that $\mu=\tilde{X}_T\tilde{\beta}_T^*$.

In Section 4.2 we infer identifiability of a true model T from Proposition 2 and the following inequality

$$\delta(T,p)\leq\delta(T). \tag{12}$$

Indeed, for any J such that $J\not\supseteq T$ and $|J|\leq|T|$ there exists $j\in T$ such that $J\subseteq F\setminus\{j\}$. Thus we obtain $\delta(T\parallel F\setminus\{j\})\leq\delta(T\parallel J)$ and minimizing both sides yields (12).

3.2 Restricted Eigenvalues

For $J\subseteq F$, $\bar{J}=F\setminus J$ and $c>0$ let

$$\kappa^2(J,c)=\min_{\nu\neq0,|\nu_{\bar{J}}|\leq c|\nu_J|}\frac{\nu^T\Sigma\nu}{\nu_J^T\nu_J}\qquad\text{and}\qquad\kappa^2(s,c)=\min_{J:|J|\leq s}\kappa(J,c).$$

Both coefficients will be called *restricted eigenvalues* of Σ. Observe that

$$\kappa^2(J,c)=\min_{\nu\neq0,|\nu_{\bar{J}}|\leq c|\nu_J|}\frac{||X_0\nu||^2}{||\nu_J||^2}=\min_{\nu\neq0,|\nu_{\bar{J}}|\leq c|\nu_J|}\frac{||X_0\nu_J-X_0\nu_{\bar{J}}||^2}{||\nu_J||^2}. \tag{13}$$

The coefficient $\kappa(s,c)$ is a modified version of an index introduced in Bickel et al. (2009). Modification consists in replacing X appearing in the original definition by X_0 and omitting the term $n^{-1/2}$. Pertaining parameters for a fixed set of predictors J and their various modifications were introduced and applied to bound the Lasso errors by van de Geer and Bühlmann (2009).

In order to study relations between sparse and restricted eigenvalues we set

$$\kappa^2(J,0)=\min_{\nu\neq0,\mathrm{supp}(\nu)\subseteq J}\frac{\nu^T\Sigma\nu}{\nu^T\nu}\qquad\text{and}\qquad\kappa^2(s,0)=\min_{J:|J|\leq s}\kappa^2(J,0).$$

Note that if X_0 is defined in (4) or in remark below (6) applies we have that $\max_{1\leq j\leq p}||x_{0j}|| \leq 1$. Thus from Rayleigh-Ritz theorem we have

$$\kappa^2(J,0) = \lambda_{min}(\Sigma_J) \leq \frac{tr(\Sigma_J)}{|J|} \leq 1 \wedge \lambda_{max}(\Sigma_J). \tag{14}$$

The upper bound above equals 1 when the columns are normalized or standardized. Note that $\kappa(J,c)$ and $\kappa(s,c)$ are nonincreasing functions of both arguments. Moreover, $\kappa^2(J,c) \leq \kappa^2(J,0)$ and $\kappa^2(s,c) \leq \kappa^2(s,0)$. This holds in view of an observation that for any fixed J and $c>0$, any ν such that $\mathrm{supp}(\nu) \subseteq J$ satisfies $\nu = \nu_J$ and thus $|\nu_{\bar{J}}| \leq c|\nu_J|$. It is easy to show also that $\kappa^2(J,c) \to \kappa^2(J,0)$ and $\kappa^2(s,c) \to \kappa^2(s,0)$ monotonically when $c \to 0^+$. Another less obvious bound, which is used in the following is stated below.

Proposition 3 *For any $s \in \mathbf{N}$ and $c>0$*

$$\kappa^2(s,c) \leq (\lfloor c \rfloor + 1)\kappa^2((\lfloor c \rfloor + 1)s, 0).$$

Condition $\kappa(s,c)>0$ imposed on matrix X is called *restricted eigenvalue condition* in Bickel et al. (2009) for their slightly different κ. Proposition 3 generalizes an observation there (p. 1720) that if the restricted eigenvalue condition holds for $c \geq 1$, then all square submatrices of Σ of size $2s$ are necessarily positive definite. Indeed, the proposition above implies that $\kappa(2s,0)>0$ from which the observation follows. Positiveness of $\kappa(T,c)$ which due to the restriction on vectors ν over which minimization is performed can hold even for $p>n$, is a certain condition on weak correlation of columns. This condition, which will be assumed later, is much less stringent than $\kappa(|T|,c)>0$, as it allows for example replication of columns belonging to the complement of T. Moreover $\kappa(T,c)>0$ for $c \geq 1$ implies identifiability of a true model.

Proposition 4 *There exists at most one true model T such that $\kappa(T,1)>0$.*

It follows that if $\kappa(T,1)>0$, then columns of X_T are linearly independent and, consequently, there exists at most one $\tilde{\beta}^*_T$ such that $\mu = \tilde{X}_T\tilde{\beta}^*_T$.

The following $\kappa - \delta$ *inequalities* follow from the Propositions 1 (ii) and the Proposition 3. We set $\theta^*_{\min} = \min_{j\in T}|\theta^*_j|$ and $t = |T|$.

Proposition 5 *We have*

$$\kappa^2(T,3)\theta^{*2}_{\min} \leq \delta(T,t) \tag{15}$$

and

$$\kappa^2(t,3)\theta^{*2}_{\min} \leq 4\delta(T,4t). \tag{16}$$

4. Error Bounds for the SOS and OS Algorithms

In this section we present the main result that is non-asymptotic bounds on the error probabilities for all steps of the SOS algorithm. The errors of consecutive steps of SOS constitute decomposition of the selection error into four parts. Two errors which can be possibly committed in the selection step correspond to two situations when the selected model is a proper subset or a superset of T.

4.1 Error Bounds for SOS

Let $\mathcal{S}_n$ be a family of models having no more than s predictors where s is defined below and $\mathcal{T}_n = \{S \in \mathcal{S}_n : S \supseteq T\}$ consists of all true models in $\mathcal{S}_n$. Observe that $|\mathcal{T}_n| = \sum_{k=0}^{s-t} \binom{p-t}{k}$. Moreover, let O_{S_1} denote a set of all correct orderings of S_1 that is orderings such that all true variables in S_1 precede the spurious ones. To simplify notation set $\delta_s = \delta(T, s)$, $\delta_t = \delta(T, t)$ and $\kappa = \kappa(T, 3)$. We also define two constants $c_1 = (3 + 6\sqrt{2})^{-1} \approx 0.087$ and $c_2 = (6+4\sqrt{2})^{-1} \approx 0.086$. We assume for the remaining part of the paper that $p \geq t+1 \geq 2$ as boundary cases are easy to analyze. Moreover, we assume the following condition which ensures that the size of S_1 defined in the first step of the SOS algorithm does not exceed n with large probability and consequently LS could be performed on data (y_0, X_{0S_1}). It states that

$$s = s(T) = t + \lfloor t^{1/2}\kappa^{-2} \rfloor \leq n. \tag{17}$$

Theorem 1 *(T1) If for some $a \in (0,1)$ $8a^{-1}\sigma^2 \log p \leq r_L^2 \leq b^2/36 \leq c_1^2 t^{-1}\kappa^4\theta_{min}^{*2}$, then*

$$P(S_1 \not\in \mathcal{T}_n) \leq \exp\left(- \frac{(1-a)r_L^2}{8\sigma^2}\right)\left(\frac{\pi r_L^2}{8\sigma^2}\right)^{-1/2}. \tag{18}$$

(T2) If for some $a \in (0,1)$ $a^{-1}\sigma^2 \log p \leq c_2(s-t+2)^{-1}\delta_s$, then

$$P(S_1 \in \mathcal{T}_n, \hat{O} \not\in O_{S_1}) \leq \frac{3}{2}\exp\left(- \frac{(1-a)c_2\delta_s}{\sigma^2}\right)\left(\frac{\pi c_2\delta_s}{\sigma^2}\right)^{-1/2}. \tag{19}$$

(T3) If for some $a \in (0,1)$ (a) $r < at^{-1}\delta_t$ and (b) $8a^{-1}\sigma^2 \log t \leq (1-a)^2\delta_t$, then

$$P(S_1 \in \mathcal{T}_n, \hat{O} \in O_{S_1}, |\hat{T}_{SOS}| < t) \leq \frac{1}{2}\exp\left(- \frac{(1-a)^3\delta_t}{8\sigma^2}\right)\left(\frac{\pi(1-a)^2\delta_t}{8\sigma^2}\right)^{-1/2}. \tag{20}$$

(T4) If for some $a \in (0,1)$ $4a^{-1}\sigma^2 \log p \leq r$, then

$$P(S_1 \in \mathcal{T}_n, \hat{O} \in O_{S_1}, |\hat{T}_{SOS}| > t) \leq \exp\left(- \frac{(1-a)r}{2\sigma^2}\right)\left(\frac{\pi r}{2\sigma^2}\right)^{-1/2}. \tag{21}$$

A regularity condition on the plan of experiment $\tilde{X}$ and the true $\tilde{\beta}^*$ induced by the assumption of Theorem 1 (T1), namely $8a^{-1}\sigma^2 \log p \leq c_1^2 t^{-1}\kappa^4\theta_{min}^{*2}$, is known as the *beta-min condition*. Its equivalent form, which is popular in the literature states that for some $a \in (0,1)$

$$\sqrt{8c_1^{-2}a^{-1}\sigma^2 t\kappa^{-4}\log p} \leq \min_{j\in T} ||H_0 x_j||\,|\beta_j^*|. \tag{22}$$

Observe that (22) implies that $\kappa > 0$, so it guarantees identifiability of T in view of Proposition 4.

Note that bounds in (T2) and (T3) as well as the bounds in Theorem 2 below can be interpreted as results analogous to the Sanov theorem in information theory on bounding probability of a non-typical event (cf. for example Cover and Thomas (2006), Section 11.4), as in view of Proposition 1 (i) δ_s may be expressed as $\min_{\beta\in B} 2\sigma^2 KL(\beta \parallel \beta^*)$ for a certain set B such that $\beta^* \not\in B$.

The first corollary provides an upper bound on a selection error of the SOS algorithm under simpler conditions. The assumption $r_L^2 = 4r$ is quite arbitrary, but results in the same lower bound for penalty and almost the same bound on error probability as in the Corollary 3 below. Note that boundary values of r_L^2 and r of order $\log p$ are allowed in Corollaries 1–3.

Corollary 1 *Assume (17) and $r_L^2 = 4r$. If for some $a \in (0, 1 - c_1)$ we have (i) $4a^{-1}\sigma^2 \log p \le r \le b^2/144 \le (c_1^2/4)at^{-1}\kappa^4\theta_{min}^{*2}$ and (ii) $r \le (4c_2/3)t^{-1/2}\kappa^2\delta_s$, then*

$$P(\hat{T}_{SOS} \neq T) \le 4\exp\left(-\frac{(1-a)r}{2\sigma^2}\right)\left(\frac{\pi r}{2\sigma^2}\right)^{-1/2}.$$

We consider now the results above under stronger conditions. We replace $\kappa = \kappa(T, 3)$ in (17) and the assumption (T1) by smaller $\kappa_t = \kappa(t, 3)$ and additionally assume the following *weak correlation condition*

$$\kappa_t^{-2} \le 3t^{1/2}, \tag{23}$$

which is weaker than a condition $\kappa_t^{-2} \le t^{1/2}$ in Theorem 1.1 in Zhou (2009, 2010). Observe that (23) is stronger than inequality (17) with κ_t instead of κ. Indeed, (23) implies in view of definition of s, that $s \le 4t$. Next, from Proposition 3 we obtain $0 < t^{-1/2}/3 \le \kappa_t^2 \le 4\kappa(4t, 0)$, but obviously $\kappa(4t, 0) = 0$ for $4t > n$, hence $4t \le n$ and $s \le n$. Moreover, we obtain from (16) that $(c_1^2/4)at^{-1}\kappa_t^4\theta_{min}^{*2} < (4c_2/3)t^{-1/2}\kappa_t^2\delta_s$ as $\delta_s \ge \delta_{4t}$ and $16c_2/(3c_1^2) \ge 1$. Hence the Corollary 1 simplifies to the following corollary.

Corollary 2 *Assume (23) and $r = r_L^2/4$. If for some $a \in (0, 1 - c_1)$ we have $16a^{-1}\sigma^2 \log p \le r_L^2 \le b^2/36 \le c_1^2 at^{-1}\kappa_t^4\theta_{min}^{*2}$, then*

$$P(\hat{T}_{SOS} \neq T) \le 4\exp\left(-\frac{(1-a)r_L^2}{8\sigma^2}\right)\left(\frac{\pi r_L^2}{8\sigma^2}\right)^{-1/2}.$$

Theorem 1 shows that the SOS algorithm is an improvement of the adaptive and the thresholded Lasso (see Zou, 2006; Huang et al., 2008; Meinshausen and Yu, 2009; Zhou, 2009, 2010; van de Geer et al., 2011) as under weaker assumptions on an experimental matrix than assumed there we obtain much stronger result, namely selection consistency. Indeed, assumptions of Theorem 1 are stated in terms of $\kappa(T, 3)$, δ_s and δ_t instead of $\kappa(t, 3)$, thus allowing for example replication of spurious predictors. Discussion of assumptions of Corollary 2 shows that the original conditions in Zhou (2009, 2010) are stronger than our conditions ensuring screening consistency of the thresholded Lasso. We stress also that our bounds are valid in both cases when the formal or the practical Lasso is used in the screening step. In Section 5 our results will be compared with a corresponding result for MCR.

4.2 Error Bounds for OS

Now we state the corresponding bounds for error probabilities of the OS algorithm in the case of $p \le n$. We recall that in the case of OS $S_1 = F$. Thus $\mathcal{S}_n = \mathcal{T}_n = \{S_1\}$ and $P(S_1 \notin \mathcal{T}_n) = 0$.

Theorem 2 *If for some* $a \in (0,1)$ $\quad a^{-1}\sigma^2 \log(t(p-t)) \le c_2\delta_p$, *then*

$$P(\hat{O} \not\in O) \le \frac{3}{2}\exp\Big(-\frac{(1-a)c_2\delta_p}{\sigma^2}\Big)\Big(\frac{\pi c_2\delta_p}{\sigma^2}\Big)^{-1/2}.$$

Moreover, (T3) and (T4) of Theorem 1 hold.

Observe that assumptions of Theorem 2 imply that $\delta_p > 0$ which guarantees uniqueness of T in view of (12).

The next corollary is analogous to Corollary 1 and provides an upper bound on a selection error of the OS algorithm under simpler conditions. This bound is more general than in Shao (1998) as we allow for greedy selection (specifically ordering of predictors), $p = p_n \to \infty$, $t = t_n \to \infty$ or GIC penalty may be of order n.

Corollary 3 *If for some* $a \in (0, 2c_2)$ $\;4a^{-1}\sigma^2 \log p \le r \le \min\big(at^{-1}\delta_t,\; 2c_2\delta_p\big)$, *then*

$$P(\hat{T}_{OS} \neq T) \le 3\exp\Big(-\frac{(1-a)r}{2\sigma^2}\Big)\Big(\frac{\pi r}{2\sigma^2}\Big)^{-1/2}.$$

It is somewhat surprising consequence of the Corollary 1–3 that, from an asymptotic point of view, the selection error of the SOS and OS algorithms, which are versions of a greedy GIC, is not greater than the selection error of a plain, exhaustive GIC. Specifically, if we define the exhaustive GIC selector by

$$\hat{T}_E = \operatorname{argmin}_{J: J\subseteq F, |J|\le p}\{R_J + |J|r\},$$

then it follows from the lower bound in (37) below, that for an arbitrary fixed index $j_0 \not\in T$ and $r > 0$ we have

$$P(\hat{T}_E \neq T) \ge P(R_{T\cup\{j_0\}} - R_T > r) \ge \frac{r}{r+\sigma^2}\exp\Big(-\frac{r}{2\sigma^2}\Big)\Big(\frac{\pi r}{2\sigma^2}\Big)^{-1/2}. \tag{24}$$

If the penalty term satisfies $\log p \ll r \ll \min(\delta_t/t,\; \delta_p)$ for $n \to \infty$, then from Corollary 3 and (24) we obtain

$$\overline{\lim_n} \log P(\hat{T}_{OS} \neq T) \le \underline{\lim_n} \log P(\hat{T}_E \neq T). \tag{25}$$

The last inequality indicates that it pays off to apply greedy algorithm in this context as a greedy search dramatically reduces ℓ_0 penalized LS without increasing its selection error.

The bounds on the selection error given in Corollaries 1–3 imply consistency of SOS and OS provided $r_n \to \infty$ and its strong consistency provided $r_n \ge c\log n$ for some $c > 2\sigma^2/(1-a)$. For boundary penalty $r_n = 4a^{-1}\sigma^2 \log p_n$ where $a \in (0, 2c_2)$, we obtain strong consistency of these algorithms if $n^{ca/(1-a)} \le p_n$ for some $c > 0.5$. Comparison of selection errors probabilities of the SOS and OS algorithms for $p < n$ requires further research.

5. Comparison of SOS and MCR

The SOS algorithm also turns out to be a competitor of iterative approaches which require minimization of more demanding LS penalized by quasiconvex functions (Fan and Li, 2001; Zou and Li, 2008; Zhang, 2010a,b; Zhang and Zhang, 2012; Huang and Zhang, 2012; Zhang, 2013; Wang et al., 2014). In this section we compare selection error bounds for SOS and *multi-stage convex relaxation* (MCR) studied in Zhang (2010b, 2013) which is the latest example of this group of algorithms. In Section 8 we compare SOS and MCR in numerical experiments.

5.1 Multi-stage Convex Relaxation Algorithm

Results in Zhang (2013) concern parametrization of the linear model without intercept given in (3). Moreover, coordinates of β are not individually penalized in MCR. In concordance with the discussion below equation (6) this corresponds to $H_0 = \mathbb{I}_n$ and $D = d\mathbb{I}_p$, where $d = \max_{1\le j\le p} ||x_j||$. Obviously,

$$X_0 = H_0 X D^{-1} = X/d, \quad y_0 = y, \quad X\beta^* = \mu = \mu_0 = X_0\theta^*, \quad H_{0J} = H_J, \quad J \subseteq F$$

and $||x_{0j}|| \le 1$. The MCR procedure finds for given $r_Z, b_Z > 0$ approximate solution of the quasiconvex minimization problem

$$\hat{\beta}^{MCR} = d^{-1}\text{argmin}_\theta\big\{||y - X_0\theta||^2 + 2r_Z\sum_{j=1}^{p}(|\theta_j| \wedge b_Z)\big\}. \tag{26}$$

As was shown in Zhang (2010b) a local minimum of (26) could be approximated by the following iterative convex minimization algorithm.

Algorithm *(MCR)*
Input: y, X and r_Z, b_Z, l.
Compute $d, \quad X_0 = X/d, \quad \bar{S} = F$
for $k = 1, 2, \ldots, l$ **do**
$\hat{\theta} = \text{argmin}_\theta\{||y - X_0\theta||^2 + 2r_Z|\theta_{\bar{S}}|\}$
$\bar{S} = \{j \in F : |\hat{\theta}_j| \le b_Z\}$
endfor
$S = F \setminus \bar{S}$
Output: $\hat{T}_{MCR} = S$, $\hat{\beta}^{MCR} = \hat{\theta}_S/d$.

Since $X_0\theta = X_{0S}\theta_S + X_{0\bar{S}}\theta_{\bar{S}}$ and $(I - H_S)X_{0S} = 0$, we obtain

$$||y - X_0\theta||^2 = ||H_S(y - X_{0\bar{S}}\theta_{\bar{S}}) - X_{0S}\theta_S||^2 + ||(I - H_S)(y - X_{0\bar{S}}\theta_{\bar{S}})||^2. \tag{27}$$

Let $\theta_S = W_S^+ Q_S^T(y - X_{0\bar{S}}\theta_{\bar{S}})$, where $X_{0S} = Q_S W_S$, Q_S is an orthogonal matrix, W_S^+ is a pseudoinverse of W_S and Q_S, W_S are computed from the QR or SVD decomposition of X_{0S}. Then θ_S is the LS solution for the response $y - X_{0\bar{S}}\theta_{\bar{S}}$ and predictors X_{0S} and the first term on the right in (27) equals 0. Thus if we set $y_\diamond = (I - H_S)y$ and $X_{\diamond\bar{S}} = (I - H_S)X_{0\bar{S}}$, then

$$||y - X_0\theta||^2 = ||(I - H_S)(y - X_{0\bar{S}}\theta_{\bar{S}})||^2 = ||y_\diamond - X_{\diamond\bar{S}}\theta_{\bar{S}})||^2.$$

It follows that for computing $\hat{\theta}$ in the MCR algorithm, we can use the Lasso and LS subroutines separately as in the following (cf. Zou and Li (2008), Algorithm 2).

Algorithm *(MCR via Lasso and LS)*
Input: y, X and r_Z, b_Z, l.
Compute d, $X_{\diamond\bar{S}} = X/d$, $y_\diamond = y$, $S = \emptyset$, $\bar{S} = F$
for $k = 1, 2, \ldots, l$ **do**
$\hat{\theta}_{\bar{S}} = \text{argmin}_{\theta_{\bar{S}}}\{||y_\diamond - X_{\diamond\bar{S}}\theta_{\bar{S}}||^2 + 2r_Z|\theta_{\bar{S}}|\}$
$\hat{\theta}_S = W_S^+ Q_S^T(y - X_{0\bar{S}}\hat{\theta}_{\bar{S}})$, where $X_{0S} = Q_S W_S$
and Q_S, W_S are computed from the QR or SVD decomposition of X_{0S}
$S = \{j \in F : |\hat{\theta}_j| > b_Z\}$, $\bar{S} = F \setminus S$
$X_{\diamond\bar{S}} = X_{0\bar{S}} - Q_S(Q_S^T X_{0\bar{S}})$, $y_\diamond = y - Q_S(Q_S^T y)$
endfor
Output: $\hat{T}_{MCR} = S$, $\hat{\beta}^{MCR} = \hat{\theta}_S/d$.

In the above algorithm $\hat{\theta}_{\bar{S}}$ is the Lasso estimator for the response $y_\diamond$ and the experimental matrix $X_{\diamond\bar{S}}$ and $\hat{\theta}_S$ is the LS estimator with the experimental matrix X_{0S} and the response equal to residuals of the Lasso fit $y - X_{0\bar{S}}\hat{\theta}_{\bar{S}}$. When one of the iterations returns S such that $|S| > n$ then the LS estimator can be calculated using the SVD decomposition instead of the QR decomposition. The above algorithm allows for usage of one of many implementations of the Lasso and is applied in our numerical experiments in Section 8.

5.2 Error Bound for MCR

In order to compare our results with selection error bounds in Zhang (2013), we restate his result using our notation. The proof of its equivalence with the original form is deferred to the Appendix. We stress that the Zhang's result holds for more general case of sub-Gaussian errors whereas we consider Gaussian errors only. Let $c_3 = 2/49$ and recalling that $\Sigma = X_0^T X_0 = d^{-2} X^T X$ and $\Sigma_J = X_{0J}^T X_{0J}$ we define *sparse eigenvalues* of Σ

$$\lambda_s = \min_{J:|J|\le s} \lambda_{min}(\Sigma_J) = \min_{\nu:supp(\nu)\le s} \frac{||X_0\nu||^2}{||\nu||^2} = \kappa^2(s,0),$$

$$\Lambda_s = \max_{J:|J|\le s} \lambda_{max}(\Sigma_J) = \max_{\nu:supp(\nu)\le s} \frac{||X_0\nu||^2}{||\nu||^2}.$$

Theorem 3 *(Zhang, 2013) Assume that there exist $s \ge 1.5t$ and $a \in (0,1)$ such that*
(i) (sparse eigenvalue condition) $\Lambda_s/\lambda_{1.5t+2s} \le 1 + s/(1.5t)$ and
(ii) $c_3^{-1}a^{-1}\sigma^2 \log p \le r_Z^2 \le b_Z^2\lambda_{1.5t+s}^2/81 \le (18)^{-2}\lambda_{1.5t+s}^2\theta_{min}^{*2}$,
then for $l > \lfloor 1.24 \ln t \rfloor + 1$ we have

$$P(\hat{T}_{MCR} \ne T) \le \exp\left(-\frac{(1-a)c_3 r_Z^2}{\sigma^2}\right)\left(\frac{\pi c_3 r_Z^2}{\sigma^2}\right)^{-1/2}.$$

Now we compare Theorem 3 with Corollary 2. Both results assume variants of the beta-min condition and bounds on (restricted or sparse) eigenvalues of Σ, namely the weak

correlation condition (23) in Corollary 2 and the sparse eigenvalue condition in Theorem 3, which is similar to *restricted isometry property* described in the Introduction. More specifically, observe that according to (14)

$$0 \le \lambda_{s'} \le \lambda_s \le \Lambda_1 = 1 \le \Lambda_s \le \Lambda_{s'} \le s' \wedge n$$

for $1 \le s < s' \le p$ and obviously $\lambda_s = 0$ for $s > n$. Then it follows from the sparse eigenvalue condition that $\lambda_{4.5t} \ge \lambda_{1.5t+2s} > 0$ and thus $4.5t \le n$ whereas the weak correlation condition stipulates that $4t \le n$. Whence the condition on correlation of predictors assumed in Theorem 3 is stronger than the corresponding assumption in the Corollary 2, moreover, Corollary 1 allows for replications of spurious predictors. However, from Proposition 3 we have $t^{-1/2}\kappa_t^2 < 4\lambda_{4t} \le 4\lambda_{3t}$ and thus for the minimal allowed $s = 1.5t$ and disregarding constants, Theorem 3 imposes weaker variant of the beta-min condition. It is worth noting that the considered algorithms as well as the error bounds assuming uniform weak correlation of predictors (Corollary 2 and Theorem 3) do not depend on n. Remaining error bounds require explicitly $s \le n$.

6. Properties of Post-model Selection Estimators

We list now several properties of post-model selection estimators which follow from the main results. Let $\hat{\mathcal{B}} = \mathcal{B}(\hat{T}, y)$ be any event defined in terms of given selector $\hat{T}$ and y and $\mathcal{B} = \mathcal{B}(T, y)$ be an analogous event pertaining to T and y. Let $\mathcal{B}^c$ and $\hat{\mathcal{B}}^c$ be complements of $\mathcal{B}$ and $\hat{\mathcal{B}}$, respectively. Observe that we have

$$P(\hat{\mathcal{B}}) \le P(\hat{\mathcal{B}}, \hat{T} = T) + P(\hat{T} \ne T) \le P(\mathcal{B}) + P(\hat{T} \ne T).$$

Analogously, $P(\hat{\mathcal{B}}^c) \le P(\mathcal{B}^c) + P(\hat{T} \ne T)$, which implies $P(\mathcal{B}) \le P(\hat{\mathcal{B}}) + P(\hat{T} \ne T)$. Both inequalities yield

$$|P(\hat{\mathcal{B}}) - P(\mathcal{B})| \le P(\hat{T} \ne T). \tag{28}$$

In particular, when $\mathcal{B} = \{G > u\}$ and $\hat{\mathcal{B}} = \{\hat{G} > u\}$ and G is some pivotal quantity then (28) implies that $P(\hat{\mathcal{B}})$ is approximated by $P(\mathcal{B})$ uniformly in u. For example, let $\hat{\tilde{\beta}}_T$ denote the LS estimator fitted on T, $h = t + 1$ for parametrization (2) and $h = t$ for parametrization (3) and define

$$f = f(T, y) = \frac{||\tilde{X}_T \hat{\tilde{\beta}}_T^{LS} - \tilde{X}_T \tilde{\beta}_T^*||^2/h}{||y - \tilde{X}_T \hat{\tilde{\beta}}_T^{LS}||^2/(n-h)}.$$

Observe that the variable f follows a Fisher-Snedecor distribution $\mathcal{F}_{h,n-h}$. Then the bound on the selection error given in Corollary 1, the assumption $\varepsilon \sim N(0, \sigma^2 \mathbb{I}_n)$ and (28) imply the following corollary.

Corollary 4 *Assume that conditions of Corollary 1 are satisfied. Then*

$$\sup_{u \in R} |P(\hat{f} \le u) - P(f \le u)| \le 4 \exp\left(- \frac{(1-a)r}{2\sigma^2} \right) \left(\frac{\pi r}{2\sigma^2} \right)^{-1/2}.$$

Note that any a priori upper bound on h in conjunction with Corollary 4 yields an approximate confidence region for $\tilde{\beta}^*_{\hat{T}}$.

Moreover, it follows from the Corollary 7 below that the Lasso estimator has the following estimation and prediction errors

Corollary 5 *Assume that conditions of Corollary 7 are satisfied. Then*

$$||X\hat{\beta} - X\beta^*|| = O_P\big(t_n^{1/2}\kappa_n^{-1}\sqrt{\log p_n}\big), \qquad |D(\hat{\beta} - \beta^*)| = O_P\big(t_n\kappa_n^{-2}\sqrt{\log p_n}\big),$$

where $\kappa_n = \kappa(T_n, 3)$.

Analogous properties of post-selection estimators are given below without proof for $\lambda_n = \lambda_{min}(\Sigma_{T_n})$.

Corollary 6 *(i) Assume that conditions of Corollary 1 are satisfied. Then*

$$||X\hat{\beta}^{SOS} - X\beta^*|| = O_P\big(t_n^{1/2}\big), \qquad |D(\hat{\beta}^{SOS} - \beta^*)| = O_P\big(t_n\lambda_n^{-1/2}\big),$$

(ii) Assume that conditions of Theorem 3 are satisfied. Then

$$||X\hat{\beta}^{MCR} - X\beta^*|| = O_P\big(t_n^{1/2}\big), \qquad |D(\hat{\beta}^{MCR} - \beta^*)| = O_P\big(t_n\lambda_n^{-1/2}\big),$$

In view of the inequality $\kappa_n^2 < \lambda_n$ it is seen that the estimation and prediction rates for the SOS and MCR post-selection estimators are better by the factor $\kappa_n^{-1}\sqrt{\log p_n}$ than the corresponding rates for the Lasso.

7. Error Bounds for the Lasso Estimator

We assume from now on that the general model (1) holds. Let $\mu_0 = H_0\mu$, $\mu_\beta = H_0X\beta = X_0\theta$ for an arbitrary $\beta \in \mathbf{R}^p$ and $\mu_{\hat{\beta}} = H_0X\hat{\beta} = X_0\hat{\theta}$. Moreover, $\Delta = \hat{\theta} - \theta = D(\hat{\beta} - \beta)$ and recall that Δ_J stands for subvector of Δ restricted to coordinates in J and $J_\beta = \mathrm{supp}(\beta) = \{j : \beta_j \neq 0\}$. Finally let $\mathcal{A} = \bigcap_{j=1}^p \{2|x_{0j}^T\varepsilon| \leq r_L\}$ and $\mathcal{A}^c$ be a complement of $\mathcal{A}$. From the Mill inequality (see the right hand side inequality in (37) below) we obtain for $Z \sim N(0,1)$

$$P(\mathcal{A}^c) \leq \sum_{j=1}^p P(2|x_{0j}^T\varepsilon| > r_L) = pP\Big(Z^2 > \frac{r_L^2}{4\sigma^2}\Big) \leq p\exp\Big(-\frac{r_L^2}{8\sigma^2}\Big)\Big(\frac{\pi r_L^2}{8\sigma^2}\Big)^{-1/2}. \qquad (29)$$

As a by-product of the proofs of the theorems above we state in this section a strengthened version of the Lasso error bounds and their consequences.

Theorem 4 *(i) On* $\mathcal{A}$ *we have*

$$||\mu_0 - \mu_{\hat{\beta}}|| \leq ||\mu_0 - \mu_\beta|| + 3r_L|J_\beta|^{1/2}\kappa^{-1}(J_\beta, 3). \qquad (30)$$

(ii) Moreover, on the set $\mathcal{A} \cap \{\beta : |\Delta| \leq 4|\Delta_J|\}$ *we have*

$$r_L|\Delta| \leq 2||\mu_0 - \mu_\beta||^2 + 8r_L^2|J_\beta|\kappa^{-2}(J_\beta, 3). \qquad (31)$$

Squaring both sides of (30) yields the following bound

$$||\mu_0 - \mu_{\hat{\beta}}||^2 \leq \Big(||\mu_0 - \mu_\beta|| + \frac{3r_L|J_\beta|^{1/2}}{\kappa(J_\beta, 3)}\Big)^2 = \inf_{a>0}(1+a)\Big(||\mu_0 - \mu_\beta||^2 + \frac{9r_L^2|J_\beta|}{a\kappa^2(J_\beta, 3)}\Big),$$

where the equality above is easily seen. Obviously $\kappa(|J_\beta|, 3) \leq \kappa(J_\beta, 3)$, hence (30) is tighter than Theorem 6.1 in Bickel et al. (2009) if we disregard a small difference in normalization of X mentioned in Section 3. Moreover, the bound above is valid for both the practical and the formal Lasso.

Let us note that as β in (30) is arbitrary, the minimum over all $\beta \in \mathbf{R}^P$ can be taken. Analogously we can minimize the right hand side of (31) over all $\beta : |\Delta| \leq 4|\Delta_J|$. Note also that if a parametric model $\mu = \tilde{X}_J\tilde{\beta}_J$ holds, then (33) below implies that indeed a condition $|\Delta| \leq 4|\Delta_J|$ is satisfied. The next corollary strengthens the ℓ_1 estimation error inequality (7.7) and the predictive inequality (7.8) in Theorem 7.2 in Bickel et al. (2009). Note that X below does not need to have normalized columns and the constant appearing in (7.7) and (7.8) in Bickel et al. (2009) is 16.

Corollary 7 *Let β be such that $\mu_0 = \mu_\beta$. Then (31) and (30) have the following form*

$$|\Delta| \leq 8r_L|J_\beta|\kappa^{-2}(J_\beta, 3) \quad \text{and} \quad ||\mu_{\hat{\beta}} - \mu_\beta||^2 \leq 9r_L^2|J_\beta|\kappa^{-2}(J_\beta, 3). \tag{32}$$

Moreover, we have on $\mathcal{A}$ the following bounds.

Corollary 8

$$||\Delta_J|| \leq 3r_L|J_\beta|^{1/2}\kappa^{-2}(J_\beta, 3) \quad \text{and} \quad |\Delta_J| \leq 3r_L|J_\beta|\kappa^{-2}(J_\beta, 3).$$

8. Simulational Study

In this section we investigate the performance of our implementation of SOS and compare it with MCR. We describe the framework of numerical experiments, discuss their results and draw conclusions. More detailed results are presented in Appendix A.4.

8.1 Description of the Experiments

We consider three models with number of potential predictors p exceeding number of observations n. The first model M_1 was analyzed in Zhang (2013). Beside it we introduce two models M_2 and M_3 which seem to fit even more to the *sparse high-dimensional* scenario $t \ll n \ll p$ and are described in Table 1, columns $1-4$. Observe that sparseness of the model measured by ratio p/t increases from 8.3 for M_1 to 100 for M_2 and to 400 for M_3. Corresponding ratios p/n are 2.5, 10 and 20, respectively. Note also that the assumptions of either Corollary 2 or Theorem 3 are not satisfied for M_1 as $4t > n$, whereas two remaining models satisfy $10t \leq n$. In all simulations the $n \times p$ matrix of experiment X with iid standard normal entries is generated and then its columns are normalized to have ℓ_2-norm equal to $\sqrt{n}$. A noise level is specified by $\sigma = 1$. For each replication of the true model, elements of β_T^* are independently generated from uniform distribution with parameters given in the column 5 of Table 1. Such layout resulted in signal to noise ratio $SNR = ||X_T\beta_T^*||/\sqrt{\mathbf{E}||\varepsilon||^2} = ||X_T\beta_T^*||/\sqrt{n}$ and it values averaged over replications are given in column 6 of Table 1.

model	t	n	p	β_T^*	SNR	SOS accuracy	MCR accuracy
M_1	30	100	250	$U(1,10)$	33	72 / 71	56 / 97
M_2	10	100	1000	$U(1,10)/2$	9.5	91 / 88	73 / 82
M_3	5	100	2000	$U(1,10)/3$	4.5	85 / 77	69 / 73

Table 1: Summary of the simulations (details explained in the text).

All computations have been performed using open source software R (see supplemental material at `http://www.mimuw.edu.pl/~pokar/Publications/`) using two frequently used Lasso implementations: `lars` (Efron et al., 2004) and `glmnet` (Friedman et al., 2010). Preliminary experiments indicated that using `lars` yields higher selection accuracies for SOS as well as for MCR than when using `glmnet`; even on grids of order 10^5 the gain in accuracy was around 10%. Moreover, for such dense grids `glmnet` was considerably slower. Thus in main numerical experiments `lars` has been used. We established that accuracy of SOS for all models is the highest when $r \approx 20$ and thus the value of r is fixed at 20. The MCR procedure is implemented via the Lasso and LS as described in Section 5.1. Similarly to Zhang (2013) we fixed number of iterations $l = 8$ for MCR. Thus compared algorithms have mutually corresponding parameters (r_L, b) and (r_Z, b_Z). As in Zhang (2013) we found optimal grid parameters for which selection accuracy is the highest one. In particular we confirmed high selection accuracy for the best parameters shown in Table 1 in Zhang (2013). Namely, the highest selection accuracy of MCR reported there is 93% for penalty and the threshold both equal 0.94 whereas we found selection accuracy 95% for both these parameters equal to 5. The difference is minor taking into account that the original penalty in Zhang (2013) corresponds in our implementation to $2r_Z/\sqrt{n} = r_Z/5$.

As a measure of performance of both algorithms we present in columns $7-8$ of Table 1 a percent of correct screening and percent of correct selection separated by the slash that is $100 \times \hat{P}(T \subseteq S)$ / $100 \times \hat{P}(\hat{T} = T)$. In simulations for the SOS algorithm, we used as a screening set $S = S_0 = \{j : |\hat{\theta}_j| > b\}$, since a double-pass screening S_1 does not lead to significant improvement of selection accuracy. Similarly, for MCR we considered as a screening set $S = \{j : |\hat{\theta}_j| > b_Z\}$ after the first iteration of the algorithm. Knowledge of both screening and selection errors allows us to estimate errors pertaining to ordering and greedy selection for SOS as well as advantage of MCR over the thresholded Lasso. Note that algorithms behave differently in that whereas for MCR probability of correct selection is larger than that of screening after the first iteration, the opposite is true for SOS. Both measures for all grid parameters are reported in Appendix A.4.

All results are based on $N = 5000$ replicates as for estimation a success probability $\pi \approx 0.75$ (corresponding crudely to our selection accuracies) in N Bernoulli experiments with prescribed error $\eta = 0.01$ and confidence level $1 - \gamma = 0.9$, we need $N \approx \pi(1-\pi)\eta^{-2}(\Phi^{-1}(1-\gamma/2))^2 \approx 5000$, where Φ^{-1} denotes the quantile function of the standard normal distribution.

8.2 Conclusions from the Experiments

Computing time of both SOS and MCR is dominated by calls to the `lars` function which is used to compute the Lasso, and as MCR uses $l = 8$ calls of this function and SOS only one, so MCR is around eight time slower than SOS.

For model M_1, MCR is substantially more precise then SOS in selecting the true subset of variables: 97% versus 71%. Recall that the highest accuracy given in Zhang (2013) is 93%. The SOS selection error is mostly due to the screening error of the Lasso as in the case of relatively large number of true predictors compared to n, the Lasso finds it difficult to filtering in all of them.

For models M_2 and M_3, SOS is more precise than MCR by approximately 5%. We note that optimal grid penalty r_L for SOS and MCR coincide whereas the threshold b is approximately twice as large for MCR as for SOS. As the results for SOS are better in these cases it turns out that thresholding the Lasso, ranking the remaining estimators and optimizing GIC in the nested family is superior to MCR iterations performed on the same initial Lasso estimator.

In conclusion, if we expect large number of genuine predictors compared to sample size, MCR is preferable, but for the sparse high-dimensional scenario SOS may be faster and more accurate.

For practical model selection we recommend the following easily achievable strategy. After performing the Lasso, we look at the paths of parameters and choose only those whose magnitude is substantially larger than others. This yields screening set S on which LS is computed, and then screened regressors are ordered according to their $|t|$ statistics from the fit. Finally we look for an 'elbow' of R_J in the nested family of the models $J \in \{\emptyset, \{j_1\}, \{j_1, j_2\}, \ldots, S\}$ which determines a cut-off point.

9. Concluding Remarks

We introduce the three-step SOS algorithm for a linear model selection. The most computationally demanding part of the method is screening of predictors by the Lasso. Ordering and greedy GIC could be computed using only two QR decompositions of X_{0S_1}. In the paper we give non-asymptotic upper bounds on error probabilities of each step of SOS in terms of the Lasso and GIC penalties (Theorem 1). As corollaries we obtain selection consistency for different (n, p) scenarios under conditions which are needed for screening consistency of the Lasso (Corollaries 1-2). The SOS algorithm is an improvement of the new version of the thresholded Lasso (Zhou, 2009, 2010) and turns out to be competitive for MCR, the latest quasiconvex penalized LS (Zhang, 2010b, 2013). The condition on correlation of predictors assumed there seems to be stronger than ours, whereas the beta-min condition may be weaker (compare discussion of Corollary 2 and Theorem 3). Theoretical comparison of SOS and MCR, in general, requires comparing λ_{3t} and $\kappa^2(T, 3)$ and remains an open problem. In simulations for the sparse high-dimensional scenario, SOS was faster and more accurate than MCR. For a traditional setting when $n > p$ we give Sanov-type bounds on error probabilities of the OS algorithm (Theorem 2). It is surprising consequence of Theorems 1-2 that the selection error of greedy GIC is asymptotically not larger than of exhaustive GIC, see formula (25). Comparison of selection errors probabilities of the SOS and OS algorithms for $p < n$ requires further research.

It is worth noticing that all results are proved for general form of the Lasso defined in (6), which encompasses two versions of the estimator: algorithm used in practice as well as its formal version.

Acknowledgments

We appreciate comments of two referees which greatly contributed to improving of the original manuscript.

Appendix A: Proofs and Supplemental Tables.

In the Appendix we provide all proofs and supplemental tables for numerical experiments.

A.1 Proofs for Section 3.

Proof of Proposition 1. We have

$$2\sigma^2 KL(\tilde{\beta}_T^*||\tilde{\beta}_J) = 2\sigma^2 \mathbf{E}_{\tilde{\beta}_T^*}\left(\frac{||y-\tilde{X}_J\tilde{\beta}_J||^2 - ||y-\tilde{X}_T\tilde{\beta}_T^*||^2}{2\sigma^2}\right) = ||\tilde{X}_T\tilde{\beta}_T^* - \tilde{X}_J\tilde{\beta}_J||^2.$$

The last expression is symmetric with respect to $\tilde{\beta}_T^*$ and $\tilde{\beta}_J$, thus $KL(\tilde{\beta}_T^*||\tilde{\beta}_J) = KL(\tilde{\beta}_J||\tilde{\beta}_T^*)$ and the second equality in (i) follows. For the proof of the first equality in (i) observe that $\delta(T||J) = \min_{\tilde{\beta}_J} ||\tilde{X}_T\tilde{\beta}_T^* - \tilde{X}_J\tilde{\beta}_J||^2$. The equality in (ii) follows from (10), the inequality there follows from Rayleigh-Ritz theorem. ■

Proof of Proposition 3. We can assume that $c \geq 1$. Consider a model J and a vector ν such that $J \supseteq \mathrm{supp}(\nu)$ and $|J| = (\lfloor c \rfloor + 1)s$ and $\kappa^2(\lfloor c \rfloor + 1)s, 0) = \nu^T\Sigma\nu/\nu^T\nu$. Sort coordinates of ν in nonincreasing order $|\nu_{j_1}| \geq |\nu_{j_2}| \ldots \geq |\nu_{j_{(\lfloor c \rfloor+1)s}}|$ and let $J_0 = \{j_1, \ldots, j_s\}$. Then we have $|J_0| = s$, $|\nu_{\bar{J}_0}| \leq \lfloor c \rfloor |\nu_{J_0}| \leq c|\nu_{J_0}|$ and $(\lfloor c \rfloor + 1)\nu_{J_0}^T\nu_{J_0} \geq \nu^T\nu$. Thus

$$\kappa^2(s,c) \leq \frac{\nu^T\Sigma\nu}{\nu_{J_0}^T\nu_{J_0}} \leq (\lfloor c \rfloor + 1)\frac{\nu^T\Sigma\nu}{\nu^T\nu} = (\lfloor c \rfloor + 1)\kappa^2((\lfloor c \rfloor + 1)s, 0)$$

and the conclusion follows. ■

Proof of Proposition 4. Assume by contradiction that there are two different true models T_1, T_2 such that $T_i = \mathrm{supp}(\beta_i) = \mathrm{supp}(\theta_i)$ for some different $\beta_i = D\theta_i$, $i = 1, 2$ and $\mu_0 = X_0\theta_1 = X_0\theta_2$. It is enough to prove that assumptions imply $\gamma(T_1, 1)\gamma(T_2, 1) = 0$, where $\gamma(J, c) = \inf\{||X_0\theta_J - X_0\theta_{\bar{J}}||, |\theta_J| = 1, |\theta_{\bar{J}}| \leq c\}$ as in view of (13) and Schwarz inequality $\kappa(J, c)/\sqrt{|J|} \leq \gamma(J, c)$. Define a vector θ with support equal to $T_1 \cup T_2$ in such a way that $\theta_{T_1\cap T_2} = \theta_{T_1\cap T_2,1} - \theta_{T_1\cap T_2,2}$, $\theta_{T_1\setminus T_2} = \theta_{T_1\setminus T_2,1}$ and $\theta_{T_2\setminus T_1} = \theta_{T_2\setminus T_1,2}$. As assumptions on T_1 and T_2 are symmetric we may assume that $|\theta_{T_1\setminus T_2}| \geq |\theta_{T_2\setminus T_1}|$ and let $\theta^o = \theta/|\theta_{T_1}|$. Then $|\theta^o_{T_1}| = 1$ and $|\theta^o_{\tilde{T}_1}| = |\theta^o_{T_2\setminus T_1}| \leq 1$. Moreover, $X\theta^o_{T_1} = X\theta^o_{\tilde{T}_1}$ which yields $\gamma(T_1, 1) = 0$. ■

Proof of Proposition 5. To prove (i) observe that (11) and (14) imply for $j \in T$

$$\kappa^2(T, 3) \leq \kappa^2(T, 0) \leq \theta_j^{*-2}\delta(T \parallel T \setminus \{j\}).$$

For (ii) we have

$$\begin{aligned}\kappa^2(t,3)/4 &\le \kappa^2(4t,0) = \min_{J:|J|\le 4t}\lambda_{min}(\Sigma_J) \le \min_{J:J\supseteq T,|J|\le 4t}\lambda_{min}(\Sigma_J)\\ &= \min_{J:J\not\supseteq T,|J\cup T|\le 4t}\lambda_{min}(\Sigma_{J\cup T}) \le \theta_{min}^{*-2}\min_{J:J\not\supseteq T,|J\cup T|\le 4t}\delta(T||J)\\ &\le \theta_{min}^{*-2}\min_{j\in T,J\supseteq T,|J|\le 4t}\delta(T||J\setminus\{j\}) = \theta_{min}^{*-2}\delta(T,4t),\end{aligned}$$

where the first inequality follows from the Proposition 3 and the third from (11). ■

A.2 Proofs for Section 6.

We now proceed to prove Theorem 4 and its corollaries. The following modified version of Lemma 1 in Bunea et al. (2007) holds.

Lemma 1 *(i) We have on $\mathcal{A}$ for an arbitrary $\beta\in\mathbf{R}^p$ and $J=\{j:\beta_j\neq 0\}$*

$$||\mu_0-\mu_{\hat\beta}||^2 + r_L|\Delta| \le ||\mu_0-\mu_\beta||^2 + 4r_L|\Delta_J|. \tag{33}$$

(ii) Moreover, we have

$$||\mu_0-\mu_{\hat\beta}||^2 \le ||\mu_0-\mu_\beta||^2 + 3r_L|\Delta_J|. \tag{34}$$

Proof. It follows from (6) that

$$||H_0(\varepsilon+\mu-X\hat\beta)||^2 + 2r_L|D\hat\beta| \le ||H_0(\varepsilon+\mu-X\beta)||^2 + 2r_L|D\beta|.$$

Equivalently, as H_0 is symmetric and idempotent, we get

$$||H_0(\mu-X\hat\beta)||^2 \le ||H_0(\mu-X\beta)||^2 + 2\varepsilon^T H_0X(\hat\beta-\beta) + 2r_L(|D\beta|-|D\hat\beta|).$$

Thus we obtain *the basic inequality*

$$||\mu_0-\mu_{\hat\beta}||^2 \le ||\mu_0-\mu_\beta||^2 + 2\varepsilon^T X_0(\hat\theta-\theta) + 2r_L(|\theta|-|\hat\theta|).$$

On $\mathcal{A}$ we have $|2\varepsilon^T X_0(\hat\theta-\theta)| \le 2\max_j|x_{0j}^T\varepsilon||\hat\theta-\theta| \le r_L|\hat\theta-\theta|$ and whence on this set

$$||\mu_0-\mu_{\hat\beta}||^2 + r_L|\hat\theta-\theta| \le ||\mu_0-\mu_\beta||^2 + 2r_L(|\hat\theta-\theta|+|\theta|-|\hat\theta|).$$

Note that for $j\notin J$ $|\hat\theta_j-\theta_j|+|\theta_j|-|\hat\theta_j|=0$ and thus

$$||\mu_0-\mu_{\hat\beta}||^2 + r_L|\hat\theta-\theta| \le ||\mu_0-\mu_\beta||^2 + 2r_L(|\hat\theta_J-\theta_J|+|\theta_J|-|\hat\theta_J|).$$

Thus (i) follows from triangle inequality and (ii) from (i) in view of $|\hat\theta_J-\theta_J|\le|\hat\theta-\theta|$. ■

Proof of Theorem 4. Proof of (i). Let $J=J_\beta$ and $\kappa=\kappa(J,3)$. We consider two cases: (a) $|\Delta|>4|\Delta_J|$ and (b) $|\Delta|\le 4|\Delta_J|$. In the case (a) it follows from (33) that stronger

inequality $||\mu_0 - \mu_{\hat{\beta}}|| \leq ||\mu_0 - \mu_\beta||$ holds. When (b) is satisfied we have $|\Delta_{\bar{J}}| \leq 3|\Delta_J|$ and it follows from the definition of κ that $\kappa^2||\Delta_J||^2 \leq ||X_0\Delta||^2 = ||\mu_{\hat{\beta}} - \mu_\beta||^2$ and thus

$$||\Delta_J|| \leq ||\mu_{\hat{\beta}} - \mu_\beta||\kappa^{-1}. \tag{35}$$

Using (35) and Jensen inequality we get

$$|\Delta_J| \leq |J|^{1/2}||\mu_{\hat{\beta}} - \mu_\beta||\kappa^{-1}. \tag{36}$$

It follows now from (34), (36) and triangle inequality that

$$||\mu_0 - \mu_{\hat{\beta}}||^2 \leq ||\mu_0 - \mu_\beta||^2 + 3r_L|J|^{1/2}\kappa^{-1}(||\mu_0 - \mu_{\hat{\beta}}|| + ||\mu_0 - \mu_\beta||)$$

and whence

$$(||\mu_0 + \mu_{\hat{\beta}}|| + ||\mu_0 - \mu_\beta||)(||\mu_0 - \mu_{\hat{\beta}}|| - ||\mu_0 - \mu_\beta||) \leq 3r_L|J|^{1/2}\kappa^{-1}(||\mu_0 - \mu_{\hat{\beta}}|| + ||\mu_0 - \mu_\beta||)$$

from which the conclusion follows.

Proof of (ii). Define $m = ||\mu_0 - \mu_\beta||$, $\hat{m} = ||\mu_0 - \mu_{\hat{\beta}}||$ and $c = 2r_L|J|^{1/2}\kappa^{-1}$. Using (33), (36) which holds provided $|\Delta| \leq 4|\Delta_J|$, and triangle inequality we get

$$\hat{m}^2 + r_L|\Delta| \leq m^2 + 2c(\hat{m} + m) \leq 2m^2 + c^2 + \hat{m}^2 + c^2,$$

from which the desired bound follows. ■

Proof of Corollary 8. The proof follows from inequality (35), (36) and the second inequality in Corollary 7. ■

A.3 Proofs for Section 4.

The next lemma states bounds on upper tail of χ^2_k distribution

Lemma 2 *Let W_k denote variable having χ^2_k distribution.(i) (Gordon, 1941 and Mill, 1926) We have for $k = 1$ and $x > 0$*

$$w_{xk}l_{xk} \leq P(W_k \geq x) \leq w_{xk}, \tag{37}$$

where $w_{xk} = e^{-x/2}(\frac{x}{2})^{k/2-1}\Gamma^{-1}(\frac{k}{2})$ and $l_{xk} = \frac{x}{x-k+2}$.
(ii) (Inglot and Ledwina, 2006) Let $k > 1$ and $x > k - 2$. Then

$$w_{xk} \leq P(W_k \geq x) \leq w_{xk}l_{xk}. \tag{38}$$

Proof. We provide the unified reasoning for both cases. For $x > 0$ and $k \in \mathbf{Z}$ let $I_k(x) = \int_x^\infty t^{(k/2)-1}e^{-t/2}\,dt$. Integration by parts yields

$$I_k(x) = 2x^{(k/2)-1}e^{-x/2} + (k-2)I_{k-2}(x). \tag{39}$$

It is easy to see that the following inequalities hold for $x > 0$ and $k \in \mathbf{Z}$

$$0 \le I_{k-2}(x) \le I_k(x)/x. \tag{40}$$

We treat cases $k = 1$ and $k > 1$ separately, as $k = 1$ is the only integer for which the second term on the RHS of (39) is negative. Dividing both sides of (39) by $2^{k/2}\Gamma(k/2)$, noting that the LHS is then $P(W_k \ge x)$ and using (40) we have for $k = 1$ and $x > 0$

$$P(W_k \ge x) \le e^{-x/2}\Big(\frac{x}{2}\Big)^{-1/2}\Gamma^{-1}\Big(\frac{1}{2}\Big)$$

and

$$P(W_k \ge x) \ge e^{-x/2}\Big(\frac{x}{2}\Big)^{-1/2}\Gamma^{-1}\Big(\frac{1}{2}\Big)\Big(1 - \frac{1}{1+x}\Big),$$

which proves (37). Analogously for $k = 2, 3, \ldots$ we obtain from (39) inequalities proved by Inglot and Ledwina (2006)

$$P(W_k \ge x) \le e^{-x/2}\Big(\frac{x}{2}\Big)^{k/2-1}\Gamma^{-1}\Big(\frac{k}{2}\Big)\Big(1 + \frac{k-2}{x-k+2}\Big)$$

for $x > k - 2$, and for $x > 0$

$$P(W_k \ge x) \ge e^{-x/2}\Big(\frac{x}{2}\Big)^{k/2-1}\Gamma^{-1}\Big(\frac{k}{2}\Big),$$

which proves (38). ■

Now we state the main lemma from which Theorems 1 and 2 follow. Let us recall that $c_1 = (3 + 6\sqrt{2})^{-1}$ and $c_2 = (6 + 4\sqrt{2})^{-1}$. Define $\mathcal{T}_n^o = \mathcal{T}_n \setminus \{T\}$ and observe that for OS algorithm we have $P(S_1 \notin \mathcal{T}_n) = 0$ and as $p \ge t + 1$, $\mathcal{T}_n = \mathcal{T}_n^o = \{F\}$, so $|\mathcal{T}_n^o| = 1$.

Lemma 3 *(T1) If $r_L^2 \le b^2/36 \le c_1^2 t^{-1}\kappa^4\theta_{min}^{*2}$, then*

$$P(S_1 \notin \mathcal{T}_n) \le p \exp\Big(-\frac{r_L^2}{8\sigma^2}\Big)\Big(\frac{\pi r_L^2}{8\sigma^2}\Big)^{-1/2}.$$

(T2) If $s \le n$, then

$$P(S_1 \in \mathcal{T}_n, \hat{O} \notin O_{S_1}) \le \frac{3}{2}|\mathcal{T}_n^o| t(s-t) \exp\Big(-\frac{c_2\delta_s}{\sigma^2}\Big)\Big(\frac{\pi c_2\delta_s}{\sigma^2}\Big)^{-1/2}.$$

(T3) If for some $a \in (0,1)$ $r \le at^{-1}\delta_t$, then

$$P(S_1 \in \mathcal{T}_n, \hat{O} \in O_{S_1}, |\hat{T}| < t) \le \frac{t}{2}\exp\Big(-\frac{(1-a)^2\delta_t}{8\sigma^2}\Big)\Big(\frac{\pi(1-a)^2\delta_t}{8\sigma^2}\Big)^{-1/2}.$$

(T4) Assume that $r/\sigma^2 \ge 2$ and $(r/\sigma^2) - \log(r/\sigma^2) \ge 2\log p$. Then

$$P(S_1 \in \mathcal{T}_n, \hat{O} \in O_{S_1}, |\hat{T}| > t) \le (p-t)(s-t)\exp\Big(-\frac{r}{2\sigma^2}\Big)\Big(\frac{\pi r}{2\sigma^2}\Big)^{-1/2}.$$

Proof. Observe that we may assume that $t > 0$ in proofs of $(T2) - (T3)$ as for $t = 0$ probabilities appearing in those parts are 0 and the conclusions are trivially satisfied.

Proof of (T1). It follows from (29) or equivalently from Lemma 2 that it is enough to prove that $\{S_1 \in \mathcal{T}_n\} \supseteq \mathcal{A}$ that is that on $\mathcal{A}$ we have

$$T \subseteq S_1 \qquad \text{and} \qquad |S_1| \leq t + \lfloor \sqrt{t}\kappa^{-2} \rfloor. \tag{41}$$

For parametric models $\mu_\beta = \mu_0$ and from (33) we have $|\Delta| \leq 4|\Delta_T|$ or equivalently $4|\Delta_{\bar{T}}| \leq 3|\Delta|$, which together with the first part of (32) yields

$$|\Delta_{\bar{T}}| \leq 6 r_L t \kappa^{-2}. \tag{42}$$

From the assumption $6r_L \leq b$ and (42) we obtain $|S_0 \setminus T| < |\Delta_{\bar{T}}|/b \leq t\kappa^{-2}$, $|S_0| < t(1+\kappa^{-2})$ and $B < b\sqrt{t(1+\kappa^{-2})}$. Using this and the first part of Corollary 8 we have $||\Delta_T|| + B < \theta^*_{min}$ or

$$||\Delta_T||^2 < (\theta^*_{min} - B)^2.$$

Indeed, from Corollary 8, the fact that $\kappa \leq 1$ and the assumption of the lemma, respectively, we have

$$\begin{aligned} ||\Delta_T|| + B &< 3 r_L t^{1/2} \kappa^{-2} + b\sqrt{t(1+\kappa^{-2})} \leq 0.5 b t^{1/2} \kappa^{-2} (1 + 2\sqrt{\kappa^4 + \kappa^2}) \\ &\leq 0.5(1 + 2\sqrt{2}) b t^{1/2} \kappa^{-2} = (6c_1)^{-1} b t^{1/2} \kappa^{-2} \leq \theta^*_{min}. \end{aligned}$$

Evidently, $|T \setminus S_1|(\theta^*_{min} - B)^2 \leq ||\Delta_T||^2 < (\theta^*_{min} - B)^2$ and thus we have $T \subseteq S_1$ on $\mathcal{A}$. But $S_1 \subseteq S_0$, hence $|S_0| \geq t$ and $B \geq b t^{1/2}$. Thus using (42) again, we have $|S_1 \setminus T| < |\Delta_{\bar{T}}|/B \leq t^{1/2}\kappa^{-2}$. Hence $|S_1 \setminus T| \leq \lfloor t^{1/2}\kappa^{-2} \rfloor$ and we obtain (41).

Proof of (T2). Let for $J_1 \in \mathcal{S}_n \setminus \mathcal{T}_n$ and $J_2 \in \mathcal{T}_n$ $W_{J_1 J_2} = \varepsilon^T(\tilde{H}_{J_1} - \tilde{H}_{J_1 \cap J_2})\varepsilon$, $\sigma^2 W_{J_2 J_1} = \varepsilon^T(\tilde{H}_{J_2} - \tilde{H}_{J_1 \cap J_2})\varepsilon$ and $\sigma Z_{J_1} = \tilde{\beta}^{*T}_T \tilde{X}^T_T (I - \tilde{H}_{J_1})\varepsilon / \sqrt{\delta_{J_1}}$, where $\delta_{J_1} = \delta(T \parallel J_1)$. Then we have that $W_{J_1 J_2} \sim \chi^2_d$, where $d \leq |J_1 \setminus J_2|$, $W_{J_2 J_1} \geq 0$ and $Z_{J_1} \sim N(0,1)$. We will use a popular decomposition of a difference between sums of squared residuals

$$\begin{aligned} R_{J_1} - R_{J_2} &= \tilde{\beta}^{*T}_T \tilde{X}^T_T (I - \tilde{H}_{J_1}) \tilde{X}_T \tilde{\beta}^*_T + 2\tilde{\beta}^{*T}_T \tilde{X}^T_T (I - \tilde{H}_{J_1})\varepsilon \\ &+ \varepsilon^T (I - \tilde{H}_{J_1})\varepsilon - \varepsilon^T (I - \tilde{H}_{J_2})\varepsilon \\ &= \delta_{J_1} + 2\sqrt{\delta_{J_1}}\sigma Z_{J_1} - \sigma^2 W_{J_1 J_2} + \sigma^2 W_{J_2 J_1} \\ &\geq \delta_{J_1}\Big(1 + \frac{2\sigma Z_{J_1}}{\sqrt{\delta_{J_1}}} - \frac{\sigma^2 W_{J_1 J_2}}{\delta_{J_1}}\Big). \end{aligned}$$

For fixed $S \in \mathcal{T}^o_n$ let $\bar{j} = S \setminus \{j\}$. Then we have from (9)

$$\begin{aligned} \{S_1 \in \mathcal{T}^o_n, \hat{O} \notin O_{S_1}\} &\subseteq \bigcup_{S \in \mathcal{T}^o_n} \bigcup_{j_1 \in T} \bigcup_{j_2 \in S \setminus T} \{R_{\bar{j}_1} \leq R_{\bar{j}_2}\} \\ &\subseteq \bigcup_{S \in \mathcal{T}^o_n} \bigcup_{j_1 \in T} \bigcup_{j_2 \in S \setminus T} \Big\{ - \frac{2\sigma Z_{\bar{j}_1}}{\sqrt{\delta_{\bar{j}_1}}} + \frac{\sigma^2 W_{\bar{j}_1 \bar{j}_2}}{\delta_{\bar{j}_1}} \geq 1 \Big\}, \end{aligned}$$

where $Z_{\bar{j}_1} \sim N(0,1)$ and $W_{\bar{j}_1\bar{j}_2} \sim \chi^2_d$, with $d \le 1$. Thus it follows that for $W = Z^2$ denoting r.v. with χ^2_1 distribution, we get

$$\begin{aligned} P(S_1 \in \mathcal{T}_n^o, \hat{O} \notin O_{S_1}) &\le \sum_{S\in\mathcal{T}_n^o}\sum_{j_1\in T}\sum_{j_2\in S\setminus T} P\Big(-\frac{2\sigma Z_{\bar{j}_1}}{\sqrt{\delta_{\bar{j}_1}}} + \frac{\sigma^2 W_{\bar{j}_1\bar{j}_2}}{\delta_{\bar{j}_1}} \ge 1\Big) \\ &\le \sum_{S\in\mathcal{T}_n^o}\sum_{j_1\in T}\sum_{j_2\in S\setminus T} \Big(P\Big(-\frac{2\sigma Z_{\bar{j}_1}}{\sqrt{\delta_{\bar{j}_1}}} \ge c\Big) + P\Big(\frac{\sigma^2 W_{\bar{j}_1\bar{j}_2}}{\delta_{\bar{j}_1}} \ge 1-c\Big)\Big) \\ &\le |\mathcal{T}_n^o| t(s-t)\Big(\frac{1}{2}P\Big(Z^2 \ge \frac{c^2\delta_s}{4\sigma^2}\Big) + P\Big(W \ge \frac{(1-c)\delta_s}{\sigma^2}\Big)\Big), \end{aligned}$$

where $j_1 \in T$ and $j_2 \in S\setminus T$ are fixed and we used $\delta_{\bar{j}_1} \ge \delta_s$. Choosing c such that $c^2/4 = 1-c$ that is $c = 1-2c_2$ in view of Lemma 2 we get the desired bound.

Proof of (T3). Reasoning as previously we have for $\bar{j} = T\setminus\{j\}$

$$\{S_1 \in \mathcal{T}_n, \hat{O} \in O_{S_1}, |\hat{T}| < t\} \subseteq \bigcup_{S\subset T}\{R_S + r|S| \le R_T + r|T|\} \subseteq \bigcup_{j\in T}\{R_{\bar{j}} \le R_T + rt\}.$$

Thus in view of Lemma 2 and the assumption $rt < a\delta_t$ we obtain

$$\begin{aligned} P(S_1 \in \mathcal{T}_n, \hat{O} \in O_{S_1}, |\hat{T}| < t) &\le \sum_{j\in T} P(R_{\bar{j}} \le R_T + rt) \\ &\le \sum_{j\in T} P\Big(-2\sigma Z_{\bar{j}} \ge \sqrt{\delta_{\bar{j}}}\Big(1 - \frac{rt}{\delta_{\bar{j}}}\Big)\Big) \\ &\le tP\Big(-2\sigma Z \ge \sqrt{\delta_t}\Big(1-\frac{rt}{\delta_t}\Big)\Big) \\ &= \frac{t}{2}P\Big(W \ge \frac{1}{4\sigma^2}\delta_t\Big(1-\frac{rt}{\delta_t}\Big)^2\Big) \\ &\le \frac{t}{2}\exp\Big(-\frac{(1-a)^2\delta_t}{8\sigma^2}\Big)\Big(\frac{\pi(1-a)^2\delta_t}{8\sigma^2}\Big)^{-1/2}. \end{aligned}$$

Proof of (T4). Observe first that for $m > 0$

$$\begin{aligned} &P(S_1 \in \mathcal{T}_n, \hat{O} \in O_{S_1}, |\hat{T}| = t+m) \\ &\le P(R_{T\cup\{j_1,\dots,j_m\}} + (t+m)r \le R_T + tr \text{ for some } j_1,\dots,j_m \in F\setminus T) \\ &\le \binom{p-t}{m} P(\sigma^2 W_m \ge mr) \le \frac{(p-t)^m}{m!} P(\sigma^2 W_m \ge mr) = B_m, \end{aligned}$$

where $W_m \sim \chi^2_m$. This follows since for any fixed $J = T\cup\{j_1,\dots,j_m\}$ we have $R_T - R_J \sim \sigma^2\chi^2_d$, where $d \le m$ and $W_d \le W_m$ in stochastic order. We will show that under conditions given in (T4) $B_m \ge B_{m+1}$ for any $m = 1,2,\dots$ thus yielding

$$P(S_1 \in \mathcal{T}_n, \hat{O} \in O_{S_1}, |\hat{T}| \ge t+m) \le (s-t-m+1)B_m,$$

which for $m = 1$ coincides with the desired inequality. Let $Q_m = B_m/B_{m+1}$, $\bar{r} = r/\sigma^2$ and observe that for $m > 1$ we have in view of (38) (note that $m\bar{r} \ge m-2$ as $\bar{r} \ge 2$)

$$Q_m \ge \frac{m+1}{p}e^{\bar{r}/2}\Big(\frac{m}{m+1}\Big)^{m/2-1}\frac{1}{\big((m+1)\bar{r}/2\big)^{1/2}}\frac{\Gamma((m+1)/2)}{\Gamma(m/2)}\frac{(m+1)\bar{r}-m+1}{(m+1)\bar{r}}.$$

Using the inequality for gamma functions (cf. formula 2.2 in Laforgia, 1984)

$$\Gamma\Big(\frac{m+1}{2}\Big)\Big/\Gamma\Big(\frac{m}{2}\Big) \geq \Big(\frac{m-1/2}{2}\Big)^{1/2}$$

we have that

$$Q_m \geq \exp\Big\{\frac{\bar{r}}{2} - \frac{1}{2}\log\bar{r} - \log p\Big\} f_1(m,\bar{r}),$$

where

$$f_1(m,\bar{r}) = \Big(\frac{m}{m+1}\Big)^{m/2-1}(m+1)^{1/2}2^{1/2}\Big(\frac{m-1/2}{2}\Big)^{1/2}\frac{(m+1)\bar{r}-m+1}{(m+1)\bar{r}}.$$

Thus in order to show that $Q_m \geq 1$ for $m > 1$ in view of assumptions it is enough to show that $f_1(m,\bar{r}) > 1$. As $f(m,\cdot)$ is increasing, it suffices to check that $f_1(m,2) > 1$. Let $f_2(m) = (\frac{m-1/2}{m+1})^{(m-1)/2}(\frac{m+3}{2})$. We have $f_1(m,2) > f_2(m)$ and $f_2(2) > 1$ thus it is enough to show that f_2 is increasing. Let

$$f_3(m) = \log(2f_2(m)) = \frac{m-1}{2}\log\frac{m-1/2}{m+1} + \log(m+3).$$

We have that

$$\begin{aligned} f_3'(m) &= \frac{1}{2}\log\frac{m-1/2}{m+1} + \frac{m-1}{2}\frac{m+1}{(m-1/2)}\frac{3}{2(m+1)^2} + \frac{1}{m+3} \\ &\geq \frac{1}{2}\frac{-3}{-3+2(m+1)} + \frac{3(m-1)}{4(m-1/2)(m+1)} + \frac{1}{m+3}, \end{aligned}$$

where the last inequality follows from $\log(1+x) > x/(1+x)$ for $x > -1$. As $1/(m+3) \geq 3/(-6+2(m+1))$ it follows that $f_3' > 0$ which implies that f_3 and thus f_2 is increasing. ■

Proof of Theorem 1. The result readily follows from Lemma 3. For (T1) we observe that

$$-\frac{r_L^2}{8\sigma^2} + \log p \leq -\frac{(1-a)r_L^2}{8\sigma^2}$$

is equivalent to $8\sigma^2 a^{-1}\log p \leq r_L^2$. Similar reasoning yields (T4). Consider derivation of (T2). From the bound

$$|\mathcal{T}_n^o| = |\mathcal{T}_n| - 1 = \sum_{k=1}^{s-t}\binom{p-t}{k} \leq (p-t) + \ldots + \frac{(p-t)^{s-t}}{(s-t)!} \leq \frac{(p-t)^{s-t}}{(s-t)!}(s-t)$$

it follows that $|\mathcal{T}_n^o|t(s-t) \leq (p-t)^{s-t}t(s-t) \leq p^{s-t}t(s-t)$. Thus the bound in (T2) will follow from $-c_2\delta_n/\sigma^2 + (s-t)\log p + \log(s-t) + \log t \leq -c_2(1-a)\delta_s/\sigma^2$ which is implied by $(s-t+2)\log p \leq c_2 a\delta_s/\sigma^2$. For (T3) we observe that

$$-\frac{(1-a)^2\delta_t}{8\sigma^2} + \log t \leq -\frac{(1-a)^3\delta_t}{8\sigma^2}$$

is equivalent to $8\sigma^2\log t \leq (1-a)^2 a\delta_t$. ■

Proof of Corollary 1. We proceed by showing that assumptions (i) and (ii) imply all assumptions of Theorem 1. We first note that (i) with the assumption $r_L^2 = 4r$ is stronger than the assumption in Theorem 1 (T1). Next, observe that condition

$$4a^{-1}\sigma^2 \log p \leq (4c_2/3)t^{-1/2}\kappa^2\delta_s \tag{43}$$

is stronger than the assumption in Theorem 1 (T2). Indeed, as $\kappa \leq 1 \leq t$ we have

$$s - t + 2 = \lfloor t^{1/2}\kappa^{-2} \rfloor + 2 \leq t^{1/2}\kappa^{-2} + 2 \leq 3t^{1/2}\kappa^{-2}.$$

Obviously, left inequalities in (i) and (ii) imply (43). Moreover, the assumption of Theorem 1 (T4) is satisfied. Furthermore, from the first $\kappa - \delta$ inequality (15) and assumption $a \in (0, 1 - c_1)$ we obtain that (i) is stronger than both conditions in Theorem 1 (T3).

In order to justify the conclusion, in view of the fact that $e^{-(1-a)x}(\pi x)^{-1/2}$ is decreasing function of $x > 0$, it is enough to show that the expressions in the exponents of the bounds (19) and (20) are larger than $r/(2\sigma^2)$ that is a value in the exponents of the bounds (18) and (21) . In the case of (19) the condition is equivalent to $r \leq 2c_2\delta_s$, which is implied by (ii). In the case of (20) the ensuing inequality is implied by $r \leq ((1-a)^2/4)\kappa^2\theta^{*2}_{min}$ which in turn is implied by (i) as $a \in (0, 1 - c_1)$. ■

Proof of Theorem 2. Let us recall that for OS algorithm we have $P(S_1 \notin \mathcal{T}_n) = 0$ and $|\mathcal{T}_n^o| = 1$, so the results follow from Lemma 3 analogously to Theorem 1. ■

Proof of Corollary 3. We proceed as in the proof of Corollary 1. The following condition

$$4a^{-1}\sigma^2 \log p \leq 2c_2\delta_s. \tag{44}$$

is stronger than the assumption in Theorem 2. The assumption imply (44) and the assumption of (T4). Furthermore, from the first $\kappa - \delta$ inequality (15) and assumption $a \in (0, 2c_2)$ we obtain that the assumption is stronger than both conditions in (T3).

Next we show that the powers in the exponents of the bounds (19) and (20) are larger than $r/(2\sigma^2)$. In the case of (19) the condition is equivalent to $r \leq 2c_2\delta_s$ which is implied by the assumption. In the case of (20) the ensuing inequality is implied by $r \leq ((1-a)^2/4)\delta_t$, which is implied by $r \leq at^{-1}\delta_t$ because for $a \in (0, 1)$ a condition $a \leq (1-a)^2/4$ is equivalent to $a \in (0, 2c_2)$. ■

A.4 Proof for Section 5.

Proof of Theorem 3. Let $v_j^T = x_j^T(I - H_T)$ for $j \notin T$ and 0 otherwise and $u_j^T = e_j^T(X_T^T X_T)^{-1}X_T^T$ for $j \in T$ and 0 otherwise, where e_j is the unit vector having 1 as the jth coordinate. Let

$$\mathcal{A} = \Big\{\forall j \in F \quad |v_j^T\varepsilon| < \frac{2r_Z}{7}\,,\, |u_j^T\varepsilon| < \frac{2r_Z}{7\lambda_t}\Big\}.$$

Using the left part of the assumption (ii), we observe that the following statement, which is equivalent of Lemma 3 in Zhang (2013) in the case of Gaussian errors, holds

$$P(\mathcal{A}^c) \leq \exp\Big(\frac{-c_3(1-a)r_Z^2}{\sigma^2}\Big)\Big(\frac{c_3\pi r_Z^2}{\sigma^2}\Big)^{-1/2}. \tag{45}$$

Then the proof of Theorem 3 follows the lines of the original proof in Zhang (2013), but just before the end we simplify the condition $l > l_0 + 1$, noting that

$$l_0 = \frac{\ln t}{2\ln(\lambda_{1.5t+s} b_Z/(6r_Z))} \leq \frac{\ln t}{2\ln(1.5)} < 1.24 \ln t.$$

In order to prove (45) observe that for $j \notin T$ $\mathrm{var}(v_j^T \varepsilon) = \sigma^2 x_j^T (I - H_T) x_j \leq \sigma^2$ and $W_j = (v_j^T \varepsilon)^2 / \mathrm{var}(v_j^T \varepsilon) \sim \chi_1^2$. Thus using Mill's inequality (37) we have

$$P\Big(|v_j^T \varepsilon| \geq \frac{2r_Z}{7}\Big) \leq P\Big(W_j \geq \frac{2c_3 r_Z^2}{\sigma^2}\Big) \leq \exp\Big(\frac{-c_3 r_Z^2}{\sigma^2}\Big)\Big(\frac{c_3 \pi r_Z^2}{\sigma^2}\Big)^{-1/2}. \tag{46}$$

Using the same reasoning for $j \in T$ with $\mathrm{var}(u_j^T \varepsilon) = \sigma^2 e_j^T (X_T^T X_T)^{-1} e_j \leq \sigma^2 \lambda_t^{-1}$ and $\tilde{W}_j = (u_j^T \varepsilon)^2 / \mathrm{var}(u_j^T \varepsilon) \sim \chi_1^2$, we have

$$P\Big(|u_j^T \varepsilon| \geq \frac{2r_L}{7\sqrt{\lambda_t}}\Big) \leq P\Big(\tilde{W}_j \geq \frac{2c_3 r_Z^2}{\sigma^2}\Big) \leq \exp\Big(\frac{-c_3 r_Z^2}{\sigma^2}\Big)\Big(\frac{c_3 \pi r_Z^2}{\sigma^2}\Big)^{-1/2}. \tag{47}$$

From (46) and (47) we obtain with $c = 2c_3 r_Z^2/\sigma^2$

$$P(\mathcal{A}^c) \leq \sum_{j \in T} P(\tilde{W}_j \geq c) + \sum_{j \notin T} P(W_j \geq c) \leq p \exp\Big(\frac{-c_3 r_Z^2}{\sigma^2}\Big)\Big(\frac{c_3 \pi r_Z^2}{\sigma^2}\Big)^{-1/2}.$$

Finally, we observe that inequality

$$-c_3 r_Z^2/\sigma^2 + \log p \leq -(1-a) c_3 r_Z^2/\sigma^2$$

is equivalent to the left part of the assumption (ii) of the theorem $c_3^{-1} a^{-1} \sigma^2 \log p \leq r_Z^2$, thus yielding (45). ■

A.4 Tables for Section 8.

$r_L \setminus b$	1.3	1.9	2.5	3.1	3.7
0.01	74.9 / 55.9	73.8 / 65.3	72.5 / 70.5	70.8 / 70.3	68.7 / 68.4
1.0	74.9 / 57.9	73.8 / 67.0	72.5 / 71.0	70.8 / 70.5	68.7 / 68.6
2.5	74.7 / 60.7	73.6 / 68.7	**72.3 / 71.3**	70.6 / 70.4	68.6 / 68.6
5.0	74.0 / 64.7	72.8 / 70.2	71.6 / 71.1	69.5 / 69.5	67.6 / 67.6
10.0	70.1 / 67.2	68.4 / 68.1	66.5 / 66.5	64.2 / 64.2	62.0 / 62.0

Table 2: Screening / selection accuracy of SOS for $M_1,\ r = 20$.

$r_L \setminus b$	0.6	0.9	1.2	1.5	1.8
5.0	95.7 / 74.0	94.2 / 78.8	92.6 / 83.9	90.6 / 86.0	88.5 / 85.7
10.0	95.5 / 78.1	94.2 / 83.4	92.5 / 86.7	90.5 / 87.1	87.8 / 85.4
15.0	94.7 / 82.0	93.1 / 85.9	**91.3 / 87.5**	89.1 / 86.6	86.1 / 84.2
20.0	93.1 / 85.1	91.2 / 86.7	89.1 / 86.4	86.1 / 84.2	83.6 / 82.2
30.0	87.6 / 84.7	85.1 / 83.1	82.2 / 80.9	78.4 / 77.4	75.2 / 74.4

Table 3: Screening / selection accuracy of SOS for $M_2,\ r = 20$.

$r_L \setminus b$	0.4	0.8	1.2	1.6	2.0
2.5	93.0 / 69.4	90.1 / 70.0	86.4 / 74.4	82.4 / 75.5	78.3 / 74.3
5.0	93.0 / 69.7	90.1 / 71.6	86.4 / 75.3	82.4 / 76.0	78.2 / 74.7
10.0	92.5 / 70.0	89.4 / 72.8	85.6 / 76.3	81.9 / 76.2	77.8 / 74.8
15.0	91.7 / 71.2	88.6 / 74.8	**84.9 / 76.9**	80.4 / 76.0	76.5 / 74.2
25.0	88.7 / 74.8	84.8 / 76.8	80.5 / 76.0	76.0 / 73.7	72.2 / 71.0
35.0	82.0 / 76.1	77.4 / 74.2	73.2 / 71.6	68.7 / 67.9	64.5 / 64.2

Table 4: Screening / selection accuracy of SOS for $M_3,\ r = 20$.

$r_Z \setminus b_Z$	4.0	5.0	6.0	7.0
0.5	67.5 / 63.0	63.3 / 90.9	57.8 / 95.6	50.7 / 94.2
2.5	67.5 / 75.0	63.3 / 94.1	57.8 / 96.3	50.7 / 94.6
5.0	66.2 / 84.5	61.9 / 95.4	**56.0 / 96.9**	48.7 / 94.8
10.0	60.8 / 90.4	55.2 / 96.5	**49.0 / 96.9**	41.8 / 94.2
20.0	43.8 / 93.9	37.3 / 96.8	31.4 / 96.0	25.6 / 90.0
30.0	28.0 / 94.2	23.0 / 95.3	18.9 / 90.0	15.1 / 78.8

Table 5: Screening / selection accuracy of MCR for $M_1,\ l = 8$.

$r_Z \setminus b_Z$	2.5	3.0	3.5	4.0
2.5	82.5 / 40.0	76.6 / 72.2	70.3 / 79.9	63.7 / 76.5
5.0	82.0 / 49.5	76.0 / 76.2	69.8 / 80.8	63.3 / 76.3
10.0	80.9 / 64.2	75.3 / 80.2	68.9 / 81.1	62.1 / 75.2
15.0	78.5 / 72.7	**72.8 / 81.9**	66.5 / 80.4	59.6 / 73.2
20.0	75.6 / 76.8	69.7 / 81.5	63.3 / 78.0	56.5 / 70.9
25.0	71.6 / 78.1	65.2 / 79.6	59.0 / 74.0	52.8 / 67.1

Table 6: Screening / selection accuracy of MCR for $M_2,\ l = 8$.

$r_Z \setminus b_Z$	1.3	1.95	2.6	3.25
5.0	85.8 / 0.2	79.0 / 28.5	71.4 / 67.1	63.1 / 68.5
10.0	85.5 / 1.6	78.0 / 45.8	70.7 / 71.1	62.4 / 68.4
15.0	84.6 / 7.7	77.4 / 58.5	**69.4 / 72.5**	61.2 / 66.9
20.0	82.9 / 22.2	75.5 / 66.9	67.1 / 72.2	59.2 / 65.0
25.0	80.2 / 40.4	72.2 / 71.4	64.5 / 70.6	56.7 / 62.6
30.0	77.1 / 53.6	69.1 / 71.2	61.2 / 67.1	54.0 / 59.3
40.0	67.1 / 64.1	60.0 / 64.3	53.1 / 58.5	46.4 / 50.9

Table 7: Screening / selection accuracy of MCR for $M_3,\ l = 8$.

References

P. Bickel, Y. Ritov, and A. Tsybakov. Simultaneous analysis of Lasso and Dantzig selector. *Annals of Statistics*, 37:1705–1732, 2009.

P. Bühlmann and S. van de Geer. *Statistics for High-dimensional Data*. Springer, New York, 2011.

F. Bunea, M. H. Wegkamp, and A. Auguste. Consistent variable selection in high dimensional regression via multiple testing. *Journal of Statistical Planning and Inference*, 136: 4349–4364, 2006.

F. Bunea, A. Tsybakov, and M. Wegkamp. Sparsity oracle inequalities for the Lasso. *Electronic Journal of Statistics*, 37:169–194, 2007.

G. Casella, F. Giron, M. Martinez, and E. Moreno. Consistency of Bayesian procedures for variable selection. *Annals of Statistics*, 37:1207–1228, 2009.

J. Chen and Z. Chen. Extended Bayesian Information Criterion for model selection with large model spaces. *Biometrika*, 95:759–771, 2008.

T. Cover and J. Thomas. *Elements of Information Theory*. Wiley, New York, 2006.

B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32:407–499, 2004.

J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96:1348–1360, 2001.

J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33:1–22, 2010.

J. Huang and C.H. Zhang. Estimation and selection via absolute penalized convex minimization and its multistage adaptive applications. *Journal of Machine Learning Research*, 13:1839–1864, 2012.

J. Huang, S. Ma, and C.H. Zhang. Adaptive Lasso for sparse high-dimensional regression models. *Statistica Sinica*, 18:1603–1618, 2008.

T. Inglot and T. Ledwina. Asymptotic optimality of new adaptive test in regression model. *Annales de l'Institut Henri Poincare. Probability and Statistics*, 42:579–590, 2006.

A. Laforgia. Further inequalities for the gamma function. *Mathematics of Computation*, 42:597–600, 1984.

S. Luo and Z. Chen. Extended BIC for linear regression models with diverging number of relevant features and high or ultra-high feature spaces. *Journal of Statistical Planning and Inference*, 143:494–504, 2013.

J. Mairal and B. Yu. Complexity analysis of the Lasso regularization path. *ArXiv*, 2012.

N. Meinshausen and P. Bühlmann. High dimensional graphs and variable selection with the Lasso. *Annals of Statistics*, 34:1436–1462, 2006.

N. Meinshausen and B. Yu. Lasso-type recovery of sparse representations for high-dimensional data. *Annals of Statistics*, 37:246–270, 2009.

B.M. Pötscher and U. Schneider. Distributional results for thresholding estimators in high-dimensional Gaussian regression models. *Electronic Journal of Statistics*, 5:1876–1934, 2011.

C.R. Rao and Y. Wu. Strongly consistent procedure for model selection in a regression problem. *Biometrika*, 76:369–374, 1989.

S. Rosset and J. Zhu. Piecewise linear regularized solution paths. *Annals of Statistics*, 35: 1012–1030, 2007.

J. Shao. Convergence rates of the Generalized Information Criterion. *Journal of Nonparametric Statistics*, 9:217–225, 1998.

R. Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society Series B*, 58:267–288, 1996.

R. Tibshirani. Regression shrinkage and selection via the Lasso: a retrospective. *Journal of the Royal Statistical Society Series B*, 73:273–282, 2011.

S. van de Geer and P. Bühlmann. On the conditions used to prove pracle results for the Lasso. *Electronic Journal of Statistics*, 3:1360–1392, 2009.

S. van de Geer, P. Bühlmann, and S. Zhou. The adaptive and the thresholded Lasso for potentially misspecified models (and a lower bound for the Lasso). *Electronic Journal of Statistics*, 5:688–749, 2011.

Z. Wang, H. Liu, and T. Zhang. Optimal computational and statistical rates of convergence for sparse nonconvex learning problems. *ArXiv*, 2014.

C.H. Zhang. Nearly unbiased variable selection under minimax concave penalty. *Annals of Statistics*, 38:894–942, 2010a.

C.H. Zhang and T. Zhang. A general theory of concave regularization for high-dimensional sparse estimation problems. *Statistical Science*, 27:576–593, 2012.

T. Zhang. Analysis of multi-stage convex relaxation for sparse regularization. *Journal of Machine Learning Research*, 11:1081–1107, 2010b.

T. Zhang. Multistage convex relaxation for feature selection. *Bernoulli*, 19:2277–2293, 2013.

P. Zhao and B. Yu. On model selection consistency of Lasso. *Journal of Machine Learning Research*, 7:2541–2563, 2006.

H. Zheng and W. Loh. Consistent variable selection in linear models. *Journal of the American Statistical Association*, 90:151–156, 1995.

S. Zhou. Thresholding procedures for high dimensional variable selection and statistical estimation. In *NIPS*, pages 2304–2312, 2009.

S. Zhou. Thresholded Lasso for high dimensional variable selection and statistical estimation. *ArXiv*, 2010.

H. Zou. The adaptive Lasso and its oracle properties. *Journal of the American Statistical Association*, 101:1418–1429, 2006.

H. Zou and R. Li. One-step sparse estimates in nonconcave penalized likelihood models. *Annals of Statistics*, 36:1509–1533, 2008.

Journal of Machine Learning Research 16 (2015) 993-1034 Submitted 12/13; Revised 8/14; Published 5/15

Learning with the Maximum Correntropy Criterion Induced Losses for Regression

Yunlong Feng YUNLONG.FENG@ESAT.KULEUVEN.BE
Xiaolin Huang HUANGXL06@MAILS.TSINGHUA.EDU.CN
Department of Electrical Engineering, ESAT-STADIUS, KU Leuven
Kasteelpark Arenberg 10, Leuven, B-3001, Belgium

Lei Shi LEISHI@FUDAN.EDU.CN
Shanghai Key Laboratory for Contemporary Applied Mathematics
School of Mathematical Sciences, Fudan University, Shanghai, 200433, P.R. China

Yuning Yang YUNING.YANG@ESAT.KULEUVEN.BE
Johan A. K. Suykens JOHAN.SUYKENS@ESAT.KULEUVEN.BE
Department of Electrical Engineering, ESAT-STADIUS, KU Leuven
Kasteelpark Arenberg 10, Leuven, B-3001, Belgium

Editor: Saharon Rosset

Abstract

Within the statistical learning framework, this paper studies the regression model associated with the correntropy induced losses. The correntropy, as a similarity measure, has been frequently employed in signal processing and pattern recognition. Motivated by its empirical successes, this paper aims at presenting some theoretical understanding towards the maximum correntropy criterion in regression problems. Our focus in this paper is two-fold: first, we are concerned with the connections between the regression model associated with the correntropy induced loss and the least squares regression model. Second, we study its convergence property. A learning theory analysis which is centered around the above two aspects is conducted. From our analysis, we see that the scale parameter in the loss function balances the convergence rates of the regression model and its robustness. We then make some efforts to sketch a general view on robust loss functions when being applied into the learning for regression problems. Numerical experiments are also implemented to verify the effectiveness of the model.

Keywords: correntropy, the maximum correntropy criterion, robust regression, robust loss function, least squares regression, statistical learning theory

1. Introduction and Motivation

Recently, a generalized correlation function named correntropy (see Santamaría et al., 2006) has drawn much attention in the signal processing and machine learning community (see Liu et al., 2007; Gunduz and Príncipe, 2009; He et al., 2011a,b). Formally speaking, correntropy is a generalized similarity measure between two scalar random variables U and V, which is defined by $\mathcal{V}_\sigma(U,V) = \mathbb{E}\mathcal{K}_\sigma(U,V)$. Here $\mathcal{K}_\sigma$ is a Gaussian kernel given by $\mathcal{K}_\sigma(u,v) = \exp\left\{-(u-v)^2/\sigma^2\right\}$ with the scale parameter $\sigma > 0$, (u,v) being a realization of (U,V).

In this paper, we are interested in the application of the similarity measure $\mathcal{V}_\sigma$ in regression problems, namely, the maximum correntropy criterion for regression. Therefore, we first assume that the data generation model is given as

$$Y = f^\star(X) + \epsilon, \quad \mathbb{E}(\epsilon \,|\, X = x) = 0. \tag{1}$$

In model (1), X is the explanatory variable that takes values in a separable metric space $\mathcal{X}$ and $Y \in \mathcal{Y} = \mathbb{R}$ stands for the response variable. The main purpose of the regression problem is to estimate f^* from a set of observations generated by (1). The underlying unknown probability distribution on $\mathcal{Z} := \mathcal{X} \times \mathcal{Y}$ is denoted as ρ.

Under the regression model (1), probably the most widely employed methodology for quantifying the regression efficiency is the mean squared error. This is the classical tool that minimizes the variance of ϵ and belongs to the second-order statistics. The drawback of the second-order statistics is that its optimality depends heavily on the assumption of Gaussianity. However, in many real-life applications, data may be contaminated by non-Gaussian noise or outliers. This motivates the introduction of the maximum correntropy criterion into the regression problems.

Given a set of i.i.d observations $\mathbf{z} = \{(x_i, y_i)\}_{i=1}^m$, for any $f : \mathcal{X} \to \mathcal{Y}$, the empirical estimator of the correntropy between $f(X)$ and Y is given as

$$\widehat{\mathcal{V}}_{\sigma,\mathbf{z}}(f) = \frac{1}{m} \sum_{i=1}^{m} \mathcal{K}_\sigma(y_i, f(x_i)).$$

The maximum correntropy criterion for regression models the output function via maximizing the empirical estimator of $\mathcal{V}_\sigma$ as follows

$$f_{\mathbf{z}} = \arg\max_{f \in \mathcal{H}} \widehat{\mathcal{V}}_{\sigma,\mathbf{z}}(f),$$

where $\mathcal{H}$ is a certain underlying hypothesis space. The maximum correntropy criterion in regression problems has shown its efficiency for cases when the noises are non-Gaussian, and also with large outliers (see Santamaría et al., 2006; Liu et al., 2007; Príncipe, 2010; Wang et al., 2013). It has also succeeded in many real-world applications, e.g., wind power forecasting (see Bessa et al., 2009) and pattern recognition (see He et al., 2011b).

In this paper, we attempt to present a theoretical understanding on the maximum correntropy criterion for regression (MCCR) within the statistical learning framework. To this end, we first generalize the idea of the maximum correntropy criterion in regression problems using the following supervised regression loss:

Definition 1 *The correntropy induced regression loss $\ell_\sigma : \mathbb{R} \times \mathbb{R} \to [0, +\infty)$ is defined as*

$$\ell_\sigma(y, t) = \sigma^2 \left(1 - e^{-\frac{(y-t)^2}{\sigma^2}}\right), \; y \in \mathcal{Y}, \, t \in \mathbb{R},$$

with $\sigma > 0$ being a scale parameter.

Figure 1 plots the correntropy induced loss function ℓ_σ (the ℓ_σ loss for short in what follows) with different choices of σ. Associated with this regression loss, the MCCR model

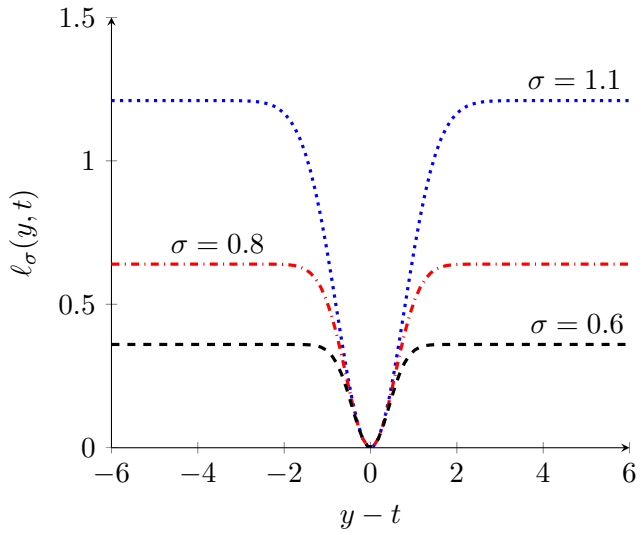


Figure 1: Plots of $\ell_\sigma(y,t) = \sigma^2(1 - e^{-(y-t)^2/\sigma^2})$ with respect to $y - t$ for different σ values: $\sigma = 0.6$ (the dashed curve), $\sigma = 0.8$ (the dotted-dashed curve), and $\sigma = 1.1$ (the dotted curve).

that we will investigate is the following empirical risk minimization (ERM) scheme

$$f_{\mathbf{z}} = \arg\min_{f\in\mathcal{H}} \frac{1}{m}\sum_{i=1}^{m} \ell_\sigma(y_i, f(x_i)), \tag{2}$$

where, throughout, the hypothesis space $\mathcal{H}$ is assumed to be a compact subset of $C(\mathcal{X})$. Here $C(\mathcal{X})$ is the Banach space of continuous functions on $\mathcal{X}$ with the norm $\|f\|_\infty = \sup_{x\in\mathcal{X}} |f(x)|$. Note that the compactness of $\mathcal{H}$ ensures the existence of the empirical target function $f_{\mathbf{z}}$.

We remark that the ℓ_σ loss is in fact a variant of the Welsch function, which was originally introduced in robust statistics (see Holland and Welsch, 1977; Dennis and Welsch, 1978). Consequently, the estimator from the MCCR model (2) is essentially a non-parametric M-estimator. For linear regression models, the robustness and the consistency properties of the M-estimator induced by the ℓ_σ loss have been investigated in Wang et al. (2013). In Santamaría et al. (2006) and Liu et al. (2007), an information-theoretical interpretation of the ℓ_σ loss by viewing it as a correlation measurement is provided.

However, existing theoretical results on understanding the ℓ_σ loss and the MCCR model are still very limited, the barriers of which lie in their non-convexity properties. From Taylor's expansion, it is easy to see that there holds $\ell_\sigma(t) \approx t^2$ for sufficiently large σ. Therefore, in some existing empirical studies, the ℓ_σ loss has been roughly taken as the least squares loss when σ is large enough. However, our studies in this paper suggest that this is in general not the case. On the other hand, the consistency property and the convergence rates of the MCCR model are yet unknown, which are the central focuses of the statistical learning research. In view of the above considerations, in this paper, our main concerns are the following two aspects:

- We are concerned with the connections between the ℓ_σ loss and the least squares loss when they are employed in regression problems. Therefore, we will study the relations between the MCCR model (2) and the ERM-based least squares regression (LSR) model.

- We are concerned with the approximation ability of the output function $f_{\mathbf{z}}$ modeled by (2). More concretely, we aim at carrying out a learning theory analysis to bound the difference between $f_{\mathbf{z}}$ and $f^\star$.

It should be mentioned that our study on the MCCR model (2) is inspired by Hu et al. (2013), which presented comprehensive and thorough studies on the minimum error entropy criterion from a learning theory viewpoint. According to Hu et al. (2013), a specific form of the minimum error entropy criterion for regression (MEECR) can be stated as

$$\widetilde{f}_{\mathbf{z}} = \arg\min_{f\in\mathcal{H}} \left\{ -\frac{\sigma^2}{m(m-1)} \sum_{i=1}^{m} \sum_{j\neq i} G\left\{ \frac{[(y_i - f(x_i)) - (y_j - f(x_j))]^2}{2\sigma^2} \right\} \right\},$$

where $G(\cdot)$ is a window function and can be chosen as $G(t) = \exp(-t)$. Hu et al. (2013, 2014) presented the first results concerning the regression consistency and convergence rates of the above MEECR model and its regularized variant when σ becomes large. Concerning the two regression models, we can see that MEECR models the empirical target function $\widetilde{f}_{\mathbf{z}}$ via a pairwise empirical risk minimization scheme while the MCCR model learns in a point-wise fashion. More discussions on the two different learning schemes will be provided in Section 2.

The rest of this paper is organized as follows. In Section 2, results on the convergence rates of the MCCR model (2) in different situations are provided. Discussions and comparisons with related studies will be also presented. Section 3 concerns connections between the two regression models: MCCR and LSR, which are explored from three aspects. Section 4 is dedicated to analyzing the MCCR model and giving proofs of theoretical results stated in Section 2. Discussions on the role that the scale parameter σ in the ℓ_σ loss plays is given in Section 5. Section 6 makes some efforts in sketching a general view of learning with robust regression losses. Numerical experiments are implemented in Section 7. We end this paper with concluding remarks in Section 8.

2. Theoretical Results on Convergence Rates and Discussions

In this section, we provide theoretical results on the convergence rates of the MCCR model (2). Explicitly, denoting $\rho_{\mathcal{X}}$ as the marginal distribution of ρ on $\mathcal{X}$, we are going to bound $\|f_{\mathbf{z}} - f_\rho\|^2_{\mathcal{L}^2_{\rho_{\mathcal{X}}}}$, where f_ρ is defined as

$$f_\rho(x) = \int_{\mathcal{Y}} y d\rho(y|x), \ \ x \in \mathcal{X},$$

and is assumed to satisfy that $f_\rho \in L^\infty_{\rho_X}$ throughout this paper. Due to the zero-mean noise assumption in the data generation model (1), almost surely there holds $f_\rho = f^\star$. To analyze

the convergence of the model, we need to introduce the following target function in $\mathcal{H}$

$$f_{\mathcal{H}} = \arg\min_{f\in\mathcal{H}} \int_{\mathcal{Z}} (f(x)-y)^2 d\rho.$$

In addition, the convergence rates that we are going to present are obtained by controlling the complexity of the hypothesis space $\mathcal{H}$. Therefore, we need the following definitions and assumptions to state our main results.

2.1 Definitions and Assumptions

Definition 2 (Covering Number) *The covering number of the hypothesis space $\mathcal{H}$, which is denoted as $\mathcal{N}(\mathcal{H},\eta)$ with the radius $\eta>0$, is defined as*

$$\mathcal{N}(\mathcal{H},\eta) := \inf\left\{ l\geq 1 : \text{there exist } f_1,\ldots,f_l\in\mathcal{H}, \text{such that } \mathcal{H}\subset\bigcup_{i=1}^{l} B(f_i,\eta)\right\},$$

where $B(f,\eta)=\{g\in\mathcal{H}:\|f-g\|_\infty\leq\eta\}$ denotes the closed ball in $C(\mathcal{X})$ with center $f\in\mathcal{H}$ and radius η.

Definition 3 (ℓ^2-Empirical Covering Number) *Let $\mathbf{x}=\{x_1,x_2,\ldots,x_n\}\subset\mathcal{X}^n$. The ℓ^2-empirical covering number of the hypothesis space $\mathcal{H}$, which is denoted as $\mathcal{N}_2(\mathcal{H},\eta)$ with radius $\eta>0$, is defined by*

$$\mathcal{N}_2(\mathcal{H},\eta) := \sup_{n\in\mathbb{N}}\sup_{\mathbf{x}\in\mathcal{X}^n}\inf\Big\{\ell\in\mathbb{N}:\exists\{f_i\}_{i=1}^{\ell}\subset\mathcal{H} \text{ such that for each } f\in\mathcal{H}, \text{there exists some } i\in\{1,2,\ldots,\ell\} \text{ with } \frac{1}{n}\sum_{j=1}^{n}|f(x_j)-f_i(x_j)|^2\leq\eta^2\Big\}.$$

Assumption 1 (Complexity Assumption I) *There exist positive constants p and $c_{I,p}$ such that*

$$\log\mathcal{N}(\mathcal{H},\eta)\leq c_{I,p}\eta^{-p},\ \forall\ \eta>0.$$

Assumption 2 (Complexity Assumption II) *There exist positive constants s and $c_{II,s}$ with $0<s<2$, such that*

$$\log\mathcal{N}_2(\mathcal{H},\eta)\leq c_{II,s}\eta^{-s},\forall\ \eta>0.$$

In learning theory, the covering number is frequently used to measure the capacity of the hypothesis spaces (see Anthony and Bartlett, 1999; Zhou, 2002). As explained in Zhou (2002), the Complexity Assumption I is typical in the statistical learning theory literature. For instance, it holds when $\mathcal{H}$ is chosen as a ball of reproducing kernel Hilbert spaces induced by Sobolev smooth kernels. The ℓ^2-empirical covering number is another data-dependent complexity measurement and usually leads to sharper convergence rates. Several examples of hypothesis spaces satisfying the Complexity Assumption II can be found in Guo and Zhou (2013).

Assumption 3 (Moment Assumption) *Assume that the tail behavior of the response variable Y satisfies $\int_{\mathcal{Z}} y^4 d\rho < \infty$.*

We will give some discussions on the above Moment Assumption in Subsection 2.3. In our study, the Moment Assumption will be employed to analyze the convergence of the MCCR model. For some specific situations of the regression model (1), in our study we will also restrict ourselves to the noise that satisfies the following Noise Assumption.

Assumption 4 (Noise Assumption) *The density function of the noise variable ϵ for any given $X = x$, which is denoted as $p_{\epsilon|X=x}$, is symmetric and uniformly bounded by the interval $[-M_0, M_0]$ with $M_0 > 0$.*

2.2 Theoretical Results on Convergence Rates

We are now ready to state our main results on the convergence rates of the MCCR model (2). Our first result considers a general case of the regression model (1), where the Moment Assumption is assumed to hold.

Theorem 4 *Assume that the Complexity Assumption I with $p > 0$ and the Moment Assumption hold. Let $f_{\mathbf{z}}$ be produced by (2). For any $0 < \delta < 1$, with confidence $1 - \delta$, there holds*

$$\|f_{\mathbf{z}} - f_\rho\|^2_{\mathcal{L}^2_{\rho_{\mathcal{X}}}} \leq 3\,\|f_{\mathcal{H}} - f_\rho\|^2_{\mathcal{L}^2_{\rho_{\mathcal{X}}}} + C_{\mathcal{H},\rho} \log(2/\delta) \left(\sigma^{-2} + \sigma m^{-1/(1+p)}\right),$$

where $C_{\mathcal{H},\rho}$ is a positive constant independent of m, σ or δ and will be given explicitly in the proof.

Discussions on the convergence rates established in Theorem 4 are postponed to Subsection 2.3. Here we remark that the moment condition in the Moment Assumption which is used in Theorem 4 can be relaxed to a weaker moment condition, i.e., $\int_{\mathcal{Z}} |y|^\ell d\rho < \infty$ with $\ell > 2$, where meaningful convergence rates can be still derived. Meanwhile, when the condition in the Moment Assumption is further strengthened, refined convergence rates can be derived. For instance, when $|y| \leq M$ almost surely for some $M > 0$, we can get the following improved convergence rates:

Theorem 5 *Assume that the Complexity Assumption II with $0 < s < 2$ holds, and $|y| \leq M$ almost surely for some $M > 0$. Let $f_\rho \in \mathcal{H}$ and $f_{\mathbf{z}}$ be produced by (2) with $\sigma = m^{1/(2+s)}$. For any $0 < \delta < 1$, with confidence $1 - \delta$, there holds*

$$\|f_{\mathbf{z}} - f_\rho\|^2_{\mathcal{L}^2_{\rho_{\mathcal{X}}}} \leq C'_{\mathcal{H},\rho} \log(2/\delta) m^{-\frac{2}{2+s}},$$

where $C'_{\mathcal{H},\rho}$ is a positive constant independent of m, σ or δ and will be given explicitly in the proof.

From Theorem 4 and Theorem 5, we can see that meaningful convergence rates can be obtained when σ is properly chosen, e.g., $\sigma = \mathcal{O}(m^\alpha)$ with some $\alpha > 0$. That is, σ has to grow in accordance with the sample size m to ensure non-trivial convergence rates. In view of this, it is natural to ask whether one can also get consistency properties or even convergence rates for the MCCR model (2) when σ is fixed. Under certain conditions, we give a positive answer in the following theorem.

Theorem 6 *Assume that the Complexity Assumption II with $0 < s < 2$ and the Noise Assumption hold. Let $f_\rho \in \mathcal{H}$, $f_{\mathbf{z}}$ be produced by (2) with σ being fixed and $\sigma > \sigma_{\mathcal{H},\rho}$ where*

$$\sigma_{\mathcal{H},\rho} = \sqrt{2}\Big(M_0 + \|f_\rho\|_\infty + \sup_{f\in\mathcal{H}} \|f\|_\infty\Big).$$

For any $0 < \delta < 1$, with confidence $1-\delta$, there holds

$$\|f_{\mathbf{z}} - f_\rho\|^2_{L^2_{\rho_\mathcal{X}}} \le C_{\mathcal{H},\sigma,\rho} \log(1/\delta) m^{-\frac{2}{2+s}},$$

where $C_{\mathcal{H},\sigma,\rho}$ is a positive constant independent of m or δ and will be given explicitly in the proof.

Proofs of the above theorems will be given in Subsection 4.3.

2.3 Discussions and Comparisons

We now give some discussions on the obtained convergence rates, the Moment Assumption and also comparisons with related studies.

2.3.1 Convergence Rates

As shown in Theorem 4, under the Moment Assumption, the convergence rates of the MCCR model depend on the choice of σ and the regularity of f_ρ. In the case when $f_\rho \in \mathcal{H}$ and $\sigma = \mathcal{O}(m^{1/(2+2p)})$, the convergence rate of $\mathcal{O}(m^{-2/(3+3p)})$ can be obtained. We then show in Theorem 5 and Theorem 6 that under the boundedness assumption on Y, or with the Noise Assumption, refined convergence rates of $\mathcal{O}(m^{-2/(2+s)})$ can be derived. Note that when s tends to zero which corresponds to the case where functions in $\mathcal{H}$ are smooth enough, convergence rates established in Theorem 5 and Theorem 6 tend to $\mathcal{O}(m^{-1})$, which are considered as the optimal rates in learning theory according to the law of large numbers (see Caponnetto and De Vito, 2007; Steinwart et al., 2009; Mendelson and Neeman, 2010; Wang and Zhou, 2011). The established convergence rates indicate the feasibility of applying the ℓ_σ loss in regression problems.

2.3.2 Moment Assumption and Related Studies on Robustness

Note that convergence rates in Theorem 4 are obtained under the Moment Assumption, which restricts the tail behavior of Y. In fact, as commented in Christmann and Steinwart (2007), tail properties of Y are frequently used in linear regression as well as nonparametric regression problems. For instance, tail behaviors of Y are usually employed to study the robustness and the consistency properties of M-estimators in linear regression problems, see e.g., Hampel et al. (1986); Davies (1993); Audibert and Catoni (2011) and many others. In the statistical learning literature, some recent studies have also confined the tail properties of Y to explore the robustness of the kernel-based regression schemes, see e.g., Christmann and Steinwart (2007); Christmann and Messem (2008); Steinwart and Christmann (2008); De Brabanter et al. (2009); Debruyne et al. (2010).

Note also that in the statistical learning literature there are many existing studies on the robust regression problem. For instance, Suykens et al. (2002a,b) presented a weighted

least squares method to pursue a robust approximation to the regression function. Debruyne et al. (2008) addressed the model selection problem in kernel-based robust regression. Some efforts have been made in Steinwart and Christmann (2008) to understand generalization abilities of regression schemes associated with convex robust loss functions, e.g., Huber's loss, which are also conducted by restricting the tail behavior of Y. As shown in Steinwart and Christmann (2008), under certain conditions, empirical estimators learned from the ERM schemes associated with certain convex robust loss functions can generalize. However, this does not directly indicate the regression consistency property of the empirical estimators, e.g., the convergence from the empirical estimator to the regression function with respect to the $\mathcal{L}^2_{\rho_\mathcal{X}}$-distance. On the other hand, as far as we can see, few studies can be found in the statistical learning literature towards understanding regression schemes associated with nonconvex robust loss functions, which are frequently employed in robust statistics (see Huber, 1981; Hampel et al., 1986).

2.3.3 Comparisons with Related Studies

As mentioned earlier, our study is motivated by recent work towards understanding the minimum error entropy criterion in regression problems (see Hu et al., 2013). Observing that when being applied to regression problems, both of the two models aim at modeling an empirical estimator that approximates the regression function f_ρ. Therefore, we can give comparisons on the convergence rates of the two models. Under the same assumptions on the tail behavior of Y and the Complexity Assumption I, when $f_\rho \in \mathcal{H}$, the convergence rates established in Hu et al. (2013) are of the type $\mathcal{O}(m^{-2/(3+3p)})$, which are presented with respect to the variance of $\widetilde{f}_\mathbf{z}(X) - f_\rho(X)$ due to the mean insensitive property of the MEECR model. In addition, when Y is bounded, under the Complexity Assumption I, Hu et al. (2013) reported convergence rates of the type $\mathcal{O}(m^{-1/(1+p)})$. In view of the convergence rates reported in Theorem 4 and Theorem 5, we conclude that the convergence rates of the two regression models are comparable. This is a nice property of the MCCR model considering that it has a lower computational complexity.

3. Connections between MCCR and LSR

As aforementioned, it is not suggested to roughly treat the ℓ_σ loss as the least squares loss in regression problems even if σ is sufficiently large. This section is dedicated to explaining this issue and trying to explore the connections between the two different regression models: MCCR and LSR.

To this end, we first give some notations. For any measurable function $f : \mathcal{X} \to \mathcal{Y}$, the generalization error of f under the ℓ_σ loss and the least squares loss are defined, respectively, as

$$\mathcal{E}^\sigma(f) = \int_\mathcal{Z} \ell_\sigma(y, f(x)) d\rho(x, y), \text{ and } \mathcal{E}(f) = \int_\mathcal{Z} (y - f(x))^2 d\rho(x, y).$$

The corresponding target functions with respect to the hypothesis space $\mathcal{H}$ are given, respectively, by

$$f^\sigma_\mathcal{H} = \arg\min_{f\in\mathcal{H}} \mathcal{E}^\sigma(f), \text{ and } f_\mathcal{H} = \arg\min_{f\in\mathcal{H}} \mathcal{E}(f).$$

3.1 A Useful Lemma

We first give a lemma which bounds the deviation of the excess risks of f associated with the ℓ_σ loss and the least squares loss for any $f \in \mathcal{H}$. It will play an important role in our following analysis. In this context, the excess risk of f with respect to the ℓ_σ loss refers to the term $\mathcal{E}^\sigma(f) - \mathcal{E}^\sigma(f_\rho)$ while the excess risk of f with respect to the least squares loss refers to the term $\mathcal{E}(f) - \mathcal{E}(f_\rho)$.

Lemma 7 *Assume that the Moment Assumption holds. For any $f \in \mathcal{H}$, the deviation of the two excess risk terms can be bounded as follows*

$$\Big| \{\mathcal{E}^\sigma(f) - \mathcal{E}^\sigma(f_\rho)\} - \{\mathcal{E}(f) - \mathcal{E}(f_\rho)\} \Big| \leq \frac{c_{\mathcal{H},\rho}}{\sigma^2},$$

where $c_{\mathcal{H},\rho}$ is a positive constant given by

$$c_{\mathcal{H},\rho} = 8 \int_{\mathcal{Z}} y^4 d\rho + 4 \sup_{f \in \mathcal{H}} \|f\|_\infty^4 + 4\|f_\rho\|_\infty^4. \tag{3}$$

Proof Following the inequality $|1 - t - e^{-t}| \leq \frac{t^2}{2}$ for $t > 0$, one has

$$\left| 1 - \frac{(y - f(x))^2}{\sigma^2} - \exp\left\{ -\frac{(y - f(x))^2}{\sigma^2} \right\} \right| \leq \frac{(y - f(x))^4}{2\sigma^4}.$$

Simple computations show that

$$\left| \mathcal{E}^\sigma(f) - \int_{\mathcal{Z}} (y - f(x))^2 d\rho \right| \leq \frac{1}{2\sigma^2} \int_{\mathcal{Z}} (y - f(x))^4 d\rho. \tag{4}$$

Since $f_\rho \in L^\infty_{\rho_X}$, the same estimation process can be applied to f_ρ, which gives

$$\left| \mathcal{E}^\sigma(f_\rho) - \int_{\mathcal{Z}} (y - f_\rho(x))^2 d\rho \right| \leq \frac{1}{2\sigma^2} \int_{\mathcal{Z}} (y - f_\rho(x))^4 d\rho. \tag{5}$$

Combining estimates in (4) and (5), we come to the following inequality

$$\Big| \{\mathcal{E}^\sigma(f) - \mathcal{E}^\sigma(f_\rho)\} - \{\mathcal{E}(f) - \mathcal{E}(f_\rho)\} \Big| \leq \frac{1}{\sigma^2} \left(8 \int_{\mathcal{Z}} y^4 d\rho + 4\|f\|_\infty^4 + 4\|f_\rho\|_\infty^4 \right),$$

where the basic inequality $(a + b)^4 \leq 8a^4 + 8b^4$ for $a, b \in \mathbb{R}$ has been applied. By denoting

$$c_{\mathcal{H},\rho} = 8 \int_{\mathcal{Z}} y^4 d\rho + 4 \sup_{f \in \mathcal{H}} \|f\|_\infty^4 + 4\|f_\rho\|_\infty^4,$$

we complete the proof of Lemma 7. ■

3.2 An Equivalence Relation between MCCR and LSR

In this part, we proceed with exploring the connections between the two models: MCCR and LSR. We will show that, when σ is large enough, under certain conditions, there does exist an equivalence relation between the two regression models. By equivalence, we mean that the two regression models admit the same target function when working in the same hypothesis space, i.e., $f_{\mathcal{H}}^{\sigma} = f_{\mathcal{H}}$ in our study.

Theorem 8 *Suppose that the Noise Assumption holds. Under the condition that $f_\rho \in \mathcal{H}$ and $\sigma > \sigma_{\mathcal{H},\rho}$ with*

$$\sigma_{\mathcal{H},\rho} = \sqrt{2}\Big(M_0 + \|f_\rho\|_\infty + \sup_{f\in\mathcal{H}} \|f\|_\infty\Big),$$

almost surely we have

$$f_{\mathcal{H}}^{\sigma} = f_{\mathcal{H}}.$$

Proof Since $f_\rho \in \mathcal{H}$, it is immediate to see that almost surely we have $f_{\mathcal{H}} = f_\rho$. To finish the proof, it remains to show that there holds $f_{\mathcal{H}}^{\sigma} = f_\rho$. In fact, for any $f \in \mathcal{H}$, we know that

$$\mathcal{E}^{\sigma}(f) = \sigma^2 \int_{\mathcal{Z}} \left(1 - \exp\left\{-\frac{(y-f(x))^2}{\sigma^2}\right\}\right) d\rho(x,y) = \sigma^2 \int_{\mathcal{X}} F_x(f(x) - f_\rho(x)) d\rho_{\mathcal{X}}(x),$$

where

$$F_x(u) := 1 - \int_{-M_0}^{M_0} \exp\left\{-\frac{(t-u)^2}{\sigma^2}\right\} p_{\epsilon|X=x}(t)dt,\ x \in \mathcal{X}.$$

By taking the derivative of F with respect to u, we get

$$F_x'(u) = -2\int_{-M_0}^{M_0} \exp\left\{-\frac{(t-u)^2}{\sigma^2}\right\} \left(\frac{t-u}{\sigma^2}\right) p_{\epsilon|X=x}(t)dt,\ x \in \mathcal{X}.$$

According to the symmetry property of $p_{\epsilon|X=x}$, we know that $F_x'(0) = 0$. Moreover,

$$F_x''(u) = 2\int_{-M_0}^{M_0} \exp\left\{-\frac{(t-u)^2}{\sigma^2}\right\} \left(\frac{\sigma^2 - 2(t-u)^2}{\sigma^4}\right) p_{\epsilon|X=x}(t)dt,\ x \in \mathcal{X}.$$

Obviously, $F_x''(u) > 0$ for all $x \in \mathcal{X}$ when $\sigma > \sigma_{\mathcal{H},\rho}$. Consequently, $u = 0$ is the unique minimizer of $F_x(\cdot)$ for any $x \in \mathcal{X}$. The proof of Theorem 8 can be completed by recalling the definitions of $f_{\mathcal{H}}^{\sigma}$ and f_ρ. ■

Theorem 8 provides a situation where the equivalence relation between the two regression models holds. In the sense of Theorem 8, one can take the ℓ_σ loss as the least squares loss when σ is large enough. However, Theorem 8 also indicates that the equivalence relation holds when the Noise Assumption is valid, $f_\rho \in \mathcal{H}$ and σ is sufficiently large. Note that the condition $f_\rho \in \mathcal{H}$ imposes a regularity requirement on the regression function f_ρ while the

Noise Assumption asks for the boundedness and symmetry of the noise. In view of these, we conclude that one is not suggested to simply treat the ℓ_σ loss as the least squares loss even if σ is sufficiently large.

We remark that Theorem 8 merely provides a sufficient condition to ensure the existence of the equivalence relation between the two models. It would be meaningful to explore some other relaxed conditions to get a similar equivalence relation. However, we also remark that the non-convexity of the ℓ_σ loss makes it non-trivial since in this case there exists more than one local optimum of the MCCR model.

3.3 Comparisons on the Convergence Rates of MCCR and LSR

To further elucidate connections between the two regression models, in this part we move our attention to comparing the learning performance of their empirical estimators, i.e., the convergence rates of $\|f_{\mathbf{z}} - f_\rho\|^2_{\mathcal{L}^2_{\rho_\mathcal{X}}}$ and $\|f^{\rm ls}_{\mathbf{z}} - f_\rho\|^2_{\mathcal{L}^2_{\rho_\mathcal{X}}}$ where $f^{\rm ls}_{\mathbf{z}}$ is modeled by the following ERM scheme

$$f^{\rm ls}_{\mathbf{z}} = \arg\min_{f\in\mathcal{H}} \frac{1}{m}\sum_{i=1}^m (f(x_i) - y_i)^2. \tag{6}$$

Noticing that due to the assumption that $\mathcal{H}$ is a compact subset of $C(\mathcal{X})$, (6) is in fact a constrained optimization model. When $\mathcal{H}$ is taken as a bounded subset of a certain reproducing kernel Hilbert space $\mathcal{H}_\mathcal{K}$, there exists an equivalence relation between the constrained optimization model (6) and the following unconstrained model

$$f^{\rm ls}_{\mathbf{z},\lambda} = \arg\min_{f\in\mathcal{H}_\mathcal{K}} \frac{1}{m}\sum_{i=1}^m (f(x_i) - y_i)^2 + \lambda\|f\|^2_\mathcal{K}, \tag{7}$$

where $\lambda > 0$ is a regularization parameter. Therefore, our comparison will be conducted between the MCCR model (2) and the regularized least squares regression model (7), which has been well understood in the statistical learning literature.

When Y is bounded, $f_\rho \in \mathcal{H}$ and the Complexity Assumption II with $0 < s < 2$ holds, the convergence rate of $\|f_{\mathbf{z}} - f_\rho\|^2_{\mathcal{L}^2_{\rho_\mathcal{X}}}$ established in Theorem 5 belongs to the type of $\mathcal{O}(m^{-2/(2+s)})$, which is the same as that of the regularized LSR (7) under the same conditions as revealed in Wu et al. (2006). In fact, when $\mathcal{H}$ is taken as a bounded subset of $\mathcal{H}_\mathcal{K}$ and the Mercer kernel $\mathcal{K}$ is sufficiently smooth, the constant s in the Complexity Assumption II can be arbitrarily small. As mentioned earlier, in this case, learning rates of the type $\mathcal{O}(m^{-1})$ can be derived which are regarded as the optimal learning rates in learning theory according to the law of large numbers.

On the other hand, due to the non-robustness of the least squares loss, almost all the existing convergence rates established for (7) are reported under the restriction that the response variable has a sub-Gaussian tail (see Wu et al., 2006; Caponnetto and De Vito, 2007; Steinwart et al., 2009; Mendelson and Neeman, 2010; Wang and Zhou, 2011). However, we see from Theorem 4 that for the MCCR model, convergence rates can be obtained under the Moment Assumption. This shows that the MCCR model can deal with non-Gaussian noise, which consequently distinguishes the two models in terms of conditions needed to establish meaningful convergence rates.

Before ending this section, let us briefly summarize the connections between MCCR and LSR as follows:

- For any given $f \in \mathcal{H}$, the difference between the excess risk of f with respect to the two regression models can be upper bounded by $\mathcal{O}(\sigma^{-2})$;

- Under certain conditions, we do see the existence of an equivalence relation between the two models, as commonly expected when σ is large enough. However, this equivalence relation might hold only under very specific conditions as suggested by our analysis;

- The MCCR model can deal with the heavy-tailed noise while the LSR model can only deal with sub-Gaussian noise. Moreover, when being restricted to cases with the bounded output or with the Gaussian noise, the performance of the two regression models are comparable. Therefore, in the above sense, we suggest that one can count on the MCCR model (2) to solve regression problems.

4. Deriving the Convergence Rates

This section presents detailed convergence analysis of the MCCR model (2) and proofs of theorems given in Section 2. The main difficulty in analyzing the model lies in the non-convexity of the loss function ℓ_σ, which disables usual techniques for analyzing convex learning models (see Cucker and Zhou, 2007; Steinwart and Christmann, 2008). We overcome this difficulty by introducing a novel error decomposition strategy with the help of Lemma 7. Analysis presented in this section is inspired by Cucker and Zhou (2007); Hu et al. (2013) and Fan et al..

4.1 Decomposing the Error into Bias-Variance Terms

The $\mathcal{L}^2_{\rho_\mathcal{X}}$-distance between the empirical target function $f_\mathbf{z}$ and the regression function f_ρ can be decomposed into the bias and the variance terms (see Vapnik, 1998; Cucker and Zhou, 2007; Steinwart and Christmann, 2008). Roughly speaking, the bias refers to the data-free error terms while the variance refers to the data-dependent error terms. The spirit of the learning theory approach to analyzing the convergence of learning models is trying to find a compromise between bias and variance by controlling the complexity of the hypothesis space. The following proposition offers a method for such compromise with respect to the MCCR model (2).

Proposition 9 *Assume that the Moment Assumption holds and let $f_\mathbf{z}$ be produced by (2). The $\mathcal{L}^2_{\rho_\mathcal{X}}$-distance between $f_\mathbf{z}$ and f_ρ can be decomposed as follows:*

$$\|f_\mathbf{z} - f_\rho\|^2_{\mathcal{L}^2_{\rho_\mathcal{X}}} \leq \mathcal{A}_{\mathcal{H},\sigma,\rho} + \mathcal{A}_{\mathcal{H},\rho} + \mathcal{S}_1(\mathbf{z}) + \mathcal{S}_2(\mathbf{z}),$$

where

$$\begin{aligned}
\mathcal{A}_{\mathcal{H},\sigma,\rho} &= 2c_{\mathcal{H},\rho}/\sigma^2, \\
\mathcal{A}_{\mathcal{H},\rho} &= \mathcal{E}(f_{\mathcal{H}}) - \mathcal{E}(f_\rho), \\
\mathcal{S}_1(\mathbf{z}) &= \{\mathcal{E}^\sigma_{\mathbf{z}}(f^\sigma_{\mathcal{H}}) - \mathcal{E}^\sigma_{\mathbf{z}}(f_\rho)\} - \{\mathcal{E}^\sigma(f^\sigma_{\mathcal{H}}) - \mathcal{E}^\sigma(f_\rho)\}, \\
\mathcal{S}_2(\mathbf{z}) &= \{\mathcal{E}^\sigma(f_{\mathbf{z}}) - \mathcal{E}^\sigma(f_\rho)\} - \{\mathcal{E}^\sigma_{\mathbf{z}}(f_{\mathbf{z}}) - \mathcal{E}^\sigma_{\mathbf{z}}(f_\rho)\}.
\end{aligned}$$

Proof Following from Lemma 7, with simple computations, we see that

$$\begin{aligned}
\|f_{\mathbf{z}} - f_\rho\|^2_{\mathcal{L}^2_{\rho_\mathcal{X}}} &\le \mathcal{E}^\sigma(f_{\mathbf{z}}) - \mathcal{E}^\sigma(f_\rho) + c_{\mathcal{H},\rho}/\sigma^2 \\
&\le \{\mathcal{E}^\sigma(f_{\mathbf{z}}) - \mathcal{E}^\sigma_{\mathbf{z}}(f_{\mathbf{z}})\} + \{\mathcal{E}^\sigma_{\mathbf{z}}(f_{\mathbf{z}}) - \mathcal{E}^\sigma_{\mathbf{z}}(f^\sigma_{\mathcal{H}})\} + \{\mathcal{E}^\sigma_{\mathbf{z}}(f^\sigma_{\mathcal{H}}) - \mathcal{E}^\sigma(f^\sigma_{\mathcal{H}})\}. \\
&\quad + \{\mathcal{E}^\sigma(f^\sigma_{\mathcal{H}}) - \mathcal{E}^\sigma(f_{\mathcal{H}})\} + \{\mathcal{E}^\sigma(f_{\mathcal{H}}) - \mathcal{E}^\sigma(f_\rho)\} + c_{\mathcal{H},\rho}/\sigma^2 \\
&\le \{\mathcal{E}^\sigma(f_{\mathbf{z}}) - \mathcal{E}^\sigma_{\mathbf{z}}(f_{\mathbf{z}})\} + \{\mathcal{E}^\sigma_{\mathbf{z}}(f_{\mathbf{z}}) - \mathcal{E}^\sigma_{\mathbf{z}}(f^\sigma_{\mathcal{H}})\} + \{\mathcal{E}^\sigma_{\mathbf{z}}(f^\sigma_{\mathcal{H}}) - \mathcal{E}^\sigma(f^\sigma_{\mathcal{H}})\}. \\
&\quad + \{\mathcal{E}^\sigma(f^\sigma_{\mathcal{H}}) - \mathcal{E}^\sigma(f_{\mathcal{H}})\} + \{\mathcal{E}(f_{\mathcal{H}}) - \mathcal{E}(f_\rho)\} + 2c_{\mathcal{H},\rho}/\sigma^2.
\end{aligned}$$

The definitions of $f_{\mathbf{z}}$, $f^\sigma_{\mathcal{H}}$ and $f_{\mathcal{H}}$ tell us that the second and the fourth terms of right-hand side of the last inequality are at most zero. By introducing intermediate terms $\mathcal{E}^\sigma_{\mathbf{z}}(f_\rho)$, $\mathcal{E}^\sigma(f_\rho)$ and corresponding notations, we finish the proof of Proposition 9. ■

As shown in Proposition 9, the $\mathcal{L}^2_{\rho_\mathcal{X}}$-distance between $f_{\mathbf{z}}$ and f_ρ are decomposed into four error terms: $\mathcal{A}_{\mathcal{H},\sigma,\rho}$, $\mathcal{A}_{\mathcal{H},\rho}$, $\mathcal{S}_1(\mathbf{z})$, and $\mathcal{S}_2(\mathbf{z})$. It is easy to see that the first two error terms are data-independent and correspond to the bias while the last two terms are data-dependent, which consequently are referred as the sample error (variance). The quantity $\mathcal{A}_{\mathcal{H},\rho}$ can be translated as the approximation ability of $f_{\mathcal{H}}$ to f_ρ, the estimation of which belongs to the topics of the approximation theory and has been well conducted. For instance, when $\mathcal{H}$ is chosen as a bounded subset of a certain reproducing kernel Hilbert space (RKHS), a comprehensive study on this term can be found in Smale and Zhou (2003). On the other hand, we remind that the bias term $\mathcal{A}_{\mathcal{H},\sigma,\rho}$ is introduced into the above error decomposition method, which not only depends on the hypothesis space $\mathcal{H}$ and the underlying probability distribution ρ, but also relies on the scale parameter σ. As explained later, this is caused by the introduction of the robustness into the regression model. This makes the decomposition strategy for the MCCR model different from those for convex regression models (see Cucker and Zhou, 2007; Steinwart and Christmann, 2008).

As a consequence of Proposition 9, to bound $\|f_{\mathbf{z}} - f_\rho\|^2_{\mathcal{L}^2_{\rho_\mathcal{X}}}$, it suffices to estimate the two sample error terms: $\mathcal{S}_1(\mathbf{z})$ and $\mathcal{S}_2(\mathbf{z})$, which will be tackled in the next subsection.

4.2 Concentration Estimates of Sample Error Terms

This part presents concentration estimates for the sample error terms $\mathcal{S}_1(\mathbf{z})$ and $\mathcal{S}_2(\mathbf{z})$ when the Moment Assumption is assumed. In learning theory, this is typically done by applying concentration inequalities to certain random variables that may be function-space valued.

In our study, for this purpose we introduce the following two random variables, $\xi_1(z)$ and $\xi_2(z)$ with $z \in \mathcal{Z}$, which are defined by

$$\xi_1(z) := -\sigma^2 \exp\left\{-(y - f^\sigma_{\mathcal{H}}(x))^2/\sigma^2\right\} + \sigma^2 \exp\left\{-(y - f_\rho(x))^2/\sigma^2\right\},$$

and

$$\xi_2(z) := -\sigma^2 \exp\left\{-(y - f_{\mathbf{z}}(x))^2/\sigma^2\right\} + \sigma^2 \exp\left\{-(y - f_\rho(x))^2/\sigma^2\right\}.$$

By applying the one-sided Bernstein's inequality in Lemma 12 to the random variable ξ_1, we can get the concentrated estimate for the sample error term $\mathcal{S}_1(\mathbf{z})$. However, the estimation of the sample error term $\mathcal{S}_2(\mathbf{z})$ requires us to apply concentration inequalities to the function-space valued random variable ξ_2 and consequently depends on the capacity of the hypothesis space $\mathcal{H}$. This is due to the fact that the random variable ξ_2 is dependent with $f_{\mathbf{z}}$ which varies in accordance with the sample $\mathbf{z}$.

Concentrated estimates for $\mathcal{S}_1(\mathbf{z})$ and $\mathcal{S}_2(\mathbf{z})$ are presented in the following two propositions, the proofs of which are given in Subsection 4.3.

Proposition 10 *Assume that the Moment Assumption holds. For any $0 < \delta < 1$, with confidence $1 - \delta/2$, there holds*

$$\mathcal{S}_1(\mathbf{z}) \leq \frac{1}{2}\left\|f_{\mathcal{H}} - f_\rho\right\|^2_{\mathcal{L}^2_{\rho_{\mathcal{X}}}} + C_{\mathcal{H},\rho,1}\left(\log\frac{2}{\delta}\right)\left(\frac{\sigma}{m} + \frac{1}{\sigma^2}\right),$$

where $C_{\mathcal{H},\rho,1}$ is a positive constant independent of m, σ or δ and will be given explicitly in the proof.

Proposition 11 *Assume that the Complexity Assumption I with $p > 0$ and the Moment Assumption hold. For any $0 < \delta < 1$, with confidence $1 - \delta/2$, there holds*

$$\mathcal{S}_2(\mathbf{z}) \leq \frac{1}{2}(\mathcal{S}_1(\mathbf{z}) + \mathcal{S}_2(\mathbf{z})) + \frac{1}{2}\left\|f_{\mathcal{H}} - f_\rho\right\|^2_{\mathcal{L}^2_{\rho_{\mathcal{X}}}} + C_{\mathcal{H},\rho,2}\left(\log\frac{2}{\delta}\right)\left\{\frac{1}{\sigma^2} + \frac{\sigma}{m^{\frac{1}{1+p}}}\right\},$$

where $C_{\mathcal{H},\rho,2}$ is a positive constant independent of m, σ or δ and will be given explicitly in the proof.

4.3 Proofs

4.3.1 Lemmas

We first list several lemmas that will be used in the proofs. Lemma 12 and Lemma 13 are one-sided Bernstein's concentration inequalities, which were introduced in Bernstein (1946) and can be found in many statistical learning textbooks, see e.g., Cucker and Zhou (2007); Steinwart and Christmann (2008). Lemma 14 was proved in Wu et al. (2007).

Lemma 12 *Let ξ be a random variable on a probability space $\mathcal{Z}$ with variance $\sigma^2_\star$ satisfying $|\xi - \mathbb{E}\xi| \leq M_\xi$ almost surely for some constant M_ξ and for all $z \in \mathcal{Z}$. Then*

$$Prob_{z\in\mathcal{Z}^m}\left\{\frac{1}{m}\sum_{i=1}^m \xi(z_i) - \mathbb{E}\xi \geq \varepsilon\right\} \leq \exp\left\{-\frac{m\varepsilon^2}{2(\sigma^2_\star + \frac{1}{3}M_\xi\varepsilon)}\right\}.$$

Lemma 13 *Let ξ be a random variable on a probability space $\mathcal{Z}$ with variance $\sigma_\star^2$ satisfying $|\xi - \mathbb{E}\xi| \leq M_\xi$ almost surely for some constant M_ξ and for all $z \in \mathcal{Z}$. Then for any $0 < \delta < 1$, with confidence $1-\delta$, we have*

$$\frac{1}{m}\sum_{i=1}^{m}\xi(z_i) - \mathbb{E}\xi \leq \frac{2M_\xi \log\frac{1}{\delta}}{3m} + \sqrt{\frac{2\sigma_\star^2 \log\frac{1}{\delta}}{m}}.$$

Lemma 14 *Let $\mathcal{F}$ be a class of measurable functions on $\mathcal{Z}$. Assume that there are constants $B, c > 0$ and $\theta \in [0,1]$ such that $\|f\|_\infty \leq B$ and $\mathbb{E}f^2 \leq c(\mathbb{E}f)^\theta$ for every $f \in \mathcal{F}$. If for some $a > 0$ and $s \in (0,2)$,*

$$\log \mathcal{N}_2(\mathcal{F}, \eta) \leq a\eta^{-s}, \qquad \forall\, \eta > 0,$$

then there exists a constant α_p depending only on p such that for any $t > 0$, with probability at least $1 - e^{-t}$, there holds

$$\mathbb{E}f - \frac{1}{m}\sum_{i=1}^{m} f(z_i) \leq \frac{1}{2}\gamma^{1-\theta}(\mathbb{E}f)^\theta + \alpha_p\gamma + 2\left(\frac{ct}{m}\right)^{\frac{1}{2-\theta}} + \frac{18Bt}{m}, \qquad \forall\, f \in \mathcal{F},$$

where

$$\gamma := \max\left\{c^{\frac{2-s}{4-2\theta+s\theta}}\left(\frac{a}{m}\right)^{\frac{2}{4-2\theta+s\theta}}, B^{\frac{2-s}{2+s}}\left(\frac{a}{m}\right)^{\frac{2}{2+s}}\right\}.$$

4.3.2 Proof of Proposition 10

Proof To bound the sample error term $\mathcal{S}_1(\mathbf{z})$, we apply the one-sided Bernstein's inequality in Lemma 13 to the random variable ξ_1 introduced in Subsection 4.2. To this end, we need to verify conditions in Lemma 13.

We first verify the boundedness condition. Recall that the random variable ξ_1 is defined as

$$\xi_1(z) := -\sigma^2 \exp\left\{-(y - f_{\mathcal{H}}^\sigma(x))^2/\sigma^2\right\} + \sigma^2 \exp\left\{-(y - f_\rho(x))^2/\sigma^2\right\},\ z \in \mathcal{Z}.$$

Introducing the auxiliary function $h(t) = \exp\{-t^2\}$ with $t \in \mathbb{R}$, it is easy to see that $\|h'\|_\infty = \sqrt{2/e}$. By taking $t_1 = (y - f_{\mathcal{H}}^\sigma(x))/\sigma$, $t_2 = (y - f_\rho(x))/\sigma$ and applying the mean value theorem to h, we see that

$$|\xi_1(z)| \leq \sqrt{2/e}\sigma|f_{\mathcal{H}}^\sigma(x) - f_\rho(x)| \leq \sqrt{2/e}\sigma\|f_{\mathcal{H}}^\sigma - f_\rho\|_\infty,\ z \in \mathcal{Z}.$$

Consequently,

$$|\xi_1 - \mathbb{E}\xi_1| \leq 2\|\xi_1\|_\infty \leq 2\sqrt{2/e}\sigma\|f_{\mathcal{H}}^\sigma - f_\rho\|_\infty \leq 2\sqrt{2/e}\sigma \sup_{f\in\mathcal{H}}\|f - f_\rho\|_\infty.$$

We are now in a position to bound the variance of the random variable ξ_1, which is denoted as $\mathrm{var}(\xi_1)$. Applying the mean value theorem to the auxiliary function $h_1(t) =$

$\exp(-t)$ at $t_1 = (y - f_{\mathcal{H}}^{\sigma}(x))^2/\sigma^2$, $t_2 = (y - f_\rho(x))^2/\sigma^2$ and recalling that $\|h_1'\|_\infty \leq 1$, we get

$$\begin{aligned}
\mathrm{var}(\xi_1) &= \mathbb{E}\xi_1^2 - (\mathbb{E}\xi_1)^2 \leq \mathbb{E}\xi_1^2 \\
&\leq \mathbb{E}\left((f_{\mathcal{H}}^{\sigma}(x) - f_\rho(x))^2 (2y - f_{\mathcal{H}}^{\sigma}(x) - f_\rho(x))^2\right) \\
&\leq \int_{\mathcal{Y}} \Big(12y^2 + 3\sup_{f\in\mathcal{H}} \|f\|_\infty^2 + 3\|f_\rho\|_\infty^2\Big) d\rho(y|x) \int_{\mathcal{X}} (f_{\mathcal{H}}^{\sigma}(x) - f_\rho(x))^2 d\rho_{\mathcal{X}}(x) \\
&= c_{\mathcal{H},\rho,0} \int_{\mathcal{X}} (f_{\mathcal{H}}^{\sigma}(x) - f_\rho(x))^2 d\rho_{\mathcal{X}}(x),
\end{aligned}$$

where the second inequality is from the elementary inequality $(a+b+c)^2 \leq 3(a^2+b^2+c^2)$ for $a, b, c \in \mathbb{R}$ and the positive constant $c_{\mathcal{H},\rho,0}$ is denoted as

$$c_{\mathcal{H},\rho,0} = 12 \int_{\mathcal{Z}} y^2 d\rho + 3 \sup_{f\in\mathcal{H}} \|f\|_\infty^2 + 3\|f_\rho\|_\infty^2. \tag{8}$$

Now applying Lemma 13 to the random variable ξ_1, we see that for any $0 < \delta < 1$, with confidence $1 - \delta/2$, there holds

$$\mathcal{S}_1(\mathbf{z}) \leq \frac{4\sqrt{2/e}\sup_{f\in\mathcal{H}} \|f - f_\rho\|_\infty}{3} \frac{\sigma \log(2/\delta)}{m} + \sqrt{\frac{2c_{\mathcal{H},\rho,0} \log(2/\delta) \|f_{\mathcal{H}}^{\sigma} - f_\rho\|_{\mathcal{L}^2_{\rho_{\mathcal{X}}}}^2}{m}}. \tag{9}$$

The elementary inequality $\sqrt{ab} \leq (a+b)/2$ for $a, b \geq 0$ gives[1]

$$\sqrt{\frac{2c_{\mathcal{H},\rho,0} \log(2/\delta) \|f_{\mathcal{H}}^{\sigma} - f_\rho\|_{\mathcal{L}^2_{\rho_{\mathcal{X}}}}^2}{m}} \leq \frac{1}{2}\|f_{\mathcal{H}}^{\sigma} - f_\rho\|_{\mathcal{L}^2_{\rho_{\mathcal{X}}}}^2 + \frac{c_{\mathcal{H},\rho,0} \log(2/\delta)}{m}. \tag{10}$$

In addition, as a consequence of Lemma 7, we have

$$\begin{aligned}
\|f_{\mathcal{H}}^{\sigma} - f_\rho\|_{\mathcal{L}^2_{\rho_{\mathcal{X}}}}^2 &\leq \mathcal{E}^{\sigma}(f_{\mathcal{H}}^{\sigma}) - \mathcal{E}^{\sigma}(f_\rho) + c_{\mathcal{H},\rho}/\sigma^2 \\
&= \mathcal{E}^{\sigma}(f_{\mathcal{H}}^{\sigma}) - \mathcal{E}^{\sigma}(f_{\mathcal{H}}) + \mathcal{E}^{\sigma}(f_{\mathcal{H}}) - \mathcal{E}^{\sigma}(f_\rho) + c_{\mathcal{H},\rho}/\sigma^2 \\
&\leq \|f_{\mathcal{H}} - f_\rho\|_{\mathcal{L}^2_{\rho_{\mathcal{X}}}}^2 + 2c_{\mathcal{H},\rho}/\sigma^2,
\end{aligned} \tag{11}$$

where the last inequality is due to the fact that $f_{\mathcal{H}}^{\sigma}$ is the minimizer of the risk functional $\mathcal{E}^{\sigma}(\cdot)$ in $\mathcal{H}$.

Combining estimates in (9), (10), and (11), we come to the conclusion that for any $0 < \delta < 1$, with confidence $1 - \delta/2$, there holds

$$\mathcal{S}_1(\mathbf{z}) \leq \frac{1}{2}\|f_{\mathcal{H}} - f_\rho\|_{\mathcal{L}^2_{\rho_{\mathcal{X}}}}^2 + C_{\mathcal{H},\rho,1} \left(\log\frac{2}{\delta}\right)\left(\frac{\sigma}{m} + \frac{1}{\sigma^2}\right),$$

where $C_{\mathcal{H},\rho,1}$ is a positive constant independent of m, σ or δ and given by

$$C_{\mathcal{H},\rho,1} = (4/3)\sqrt{2/e}\sup_{f\in\mathcal{H}} \|f - f_\rho\|_\infty + 2c_{\mathcal{H},\rho} + c_{\mathcal{H},\rho,0}.$$

Thus we have completed the proof of Proposition 10. ■

1. Refined estimate can be derived here by applying Young's inequality $ab \leq \frac{ta^2}{2} + \frac{b^2}{2t}$ for $a, b \in \mathbb{R}$, $t > 0$. In our proof, we choose $t = 1$ for simplification.

4.3.3 Proof of Proposition 11

To prove Proposition 11, we first need to prove the following intermediate conclusion, which is in fact a concentrated estimate for function-space valued random variables.

Proposition 15 *Assume that the Moment Assumption holds. Let ε satisfy $\varepsilon \geq c_{\mathcal{H},\rho}/\sigma^2$. For any $0 < \delta < 1$, with confidence $1 - \delta/2$, there holds*

$$\begin{aligned}&\mathrm{Prob}_{\mathbf{z}\in\mathcal{Z}^m}\left\{\sup_{f\in\mathcal{H}}\frac{(\mathcal{E}^\sigma(f)-\mathcal{E}^\sigma(f_\rho))-(\mathcal{E}^\sigma_{\mathbf{z}}(f)-\mathcal{E}^\sigma_{\mathbf{z}}(f_\rho))}{\sqrt{\mathcal{E}^\sigma(f)-\mathcal{E}^\sigma(f_\rho)+2\varepsilon}}>4\sqrt{\varepsilon}\right\}\\&\leq\mathcal{N}\left(\mathcal{H},\frac{\varepsilon}{\sqrt{2/e}\sigma}\right)\exp\left\{-\frac{3m\varepsilon}{4\sqrt{2/e}\sup_{f\in\mathcal{H}}\|f-f_\rho\|_\infty\sigma+6c_{\mathcal{H},\rho,0}}\right\},\end{aligned}$$

where $c_{\mathcal{H},\rho}$ is given in (3) *and $c_{\mathcal{H},\rho,0}$ is given in* (8)*, both of which are positive constants independent of m, σ or δ.*

Proof To derive the desired estimate, we will apply the one-sided Bernstein's inequality in Lemma 13 to the function set $\mathcal{H}$ by taking its capacity into account.

For any $f \in \mathcal{H}$, we redefine the random variable $\xi_2(z)$ as follows

$$\xi_2(z) = -\sigma^2\exp\left\{-(y-f(x))^2/\sigma^2\right\}+\sigma^2\exp\left\{-(y-f_\rho(x))^2/\sigma^2\right\},\ z\in\mathcal{Z}.$$

Following from the proof of Proposition 10, we know that

$$\|\xi_2\|_\infty \leq \sqrt{2/e}\sigma\|f-f_\rho\|_\infty \text{ and } |\xi_2-\mathbb{E}\xi_2|\leq 2\sqrt{2/e}\sigma\sup_{f\in\mathcal{H}}\|f-f_\rho\|_\infty.$$

Meanwhile, we also know from the proof of Proposition 10 that

$$\mathbb{E}\xi_2^2 \leq c_{\mathcal{H},\rho,0}\|f-f_\rho\|^2_{\mathcal{L}^2_{\rho_\mathcal{X}}},$$

where the constant $c_{\mathcal{H},\rho,0}$ is given in (8).

Consider a function set $\{f_j\}_{j=1}^J \subset \mathcal{H}$ with $J = \mathcal{N}(\mathcal{H}, \varepsilon/(\sqrt{2/e}\sigma))$. The compactness of $\mathcal{H}$ ensures the existence and finiteness of J. Now we let

$$\mu = \sqrt{\mathcal{E}^\sigma(f_j)-\mathcal{E}^\sigma(f_\rho)+2\varepsilon},$$

and choose ε such that $\varepsilon \geq c_{\mathcal{H},\rho}/\sigma^2$. Applying the one-sided Bernstein's inequality in Lemma 12 to the following group of random variables

$$\xi_{2,j}(z) = -\sigma^2\exp\left\{-(y-f_j(x))^2/\sigma^2\right\}+\sigma^2\exp\left\{-(y-f_\rho(x))^2/\sigma^2\right\},\ j=1,\ldots,J,$$

we come to the following conclusion

$$\begin{aligned}
&\mathrm{Prob}_{\mathbf{z}\in\mathcal{Z}^m}\left\{\frac{(\mathcal{E}^\sigma(f_j)-\mathcal{E}^\sigma(f_\rho))-(\mathcal{E}^\sigma_{\mathbf{z}}(f_j)-\mathcal{E}^\sigma_{\mathbf{z}}(f_\rho))}{\sqrt{\mathcal{E}^\sigma(f_j)-\mathcal{E}^\sigma(f_\rho)+2\varepsilon}}>\sqrt{\varepsilon}\right\}\\
&\quad\le\exp\left\{-\frac{3m\varepsilon\mu^2}{4\sqrt{2/e}\|f_j-f_\rho\|_\infty\sqrt{\varepsilon}\mu\sigma+6c_{\mathcal{H},\rho,0}\|f_j-f_\rho\|^2_{\mathcal{L}^2_{\rho_\mathcal{X}}}}\right\}\\
&\quad\le\exp\left\{-\frac{3m\varepsilon\mu^2}{4\sqrt{2/e}\|f_j-f_\rho\|_\infty\sqrt{\varepsilon}\mu\sigma+6c_{\mathcal{H},\rho,0}\mu^2}\right\}\\
&\quad\le\exp\left\{-\frac{3m\varepsilon}{4\sqrt{2/e}\sup_{f\in\mathcal{H}}\|f-f_\rho\|_\infty\sigma+6c_{\mathcal{H},\rho,0}}\right\},
\end{aligned}$$

where the last two inequalities follow from the inequality in Lemma 7, the equation that $\mathcal{E}(f_j)-\mathcal{E}(f_\rho)=\|f_j-f_\rho\|^2_{\mathcal{L}^2_{\rho_\mathcal{X}}}$, the fact that $\varepsilon\ge c_{\mathcal{H},\rho}/\sigma^2$ and

$$\mu^2=\mathcal{E}^\sigma(f_j)-\mathcal{E}^\sigma(f_\rho)+2\varepsilon\ge\mathcal{E}^\sigma(f_j)-\mathcal{E}^\sigma(f_\rho)+c_{\mathcal{H},\rho}/\sigma^2+\varepsilon\ge\mathcal{E}(f_j)-\mathcal{E}(f_\rho)+\varepsilon\ge\varepsilon.$$

From the choice of f_j, we know that for each $f\in\mathcal{H}$, there exists some j such that $\|f-f_j\|_\infty\le\varepsilon/(\sqrt{2/e}\sigma)$. Therefore $|\mathcal{E}^\sigma(f)-\mathcal{E}^\sigma(f_j)|$ and $|\mathcal{E}^\sigma_{\mathbf{z}}(f)-\mathcal{E}^\sigma_{\mathbf{z}}(f_j)|$ can be both upper bounded by ε, which yields

$$\frac{|(\mathcal{E}^\sigma_{\mathbf{z}}(f)-\mathcal{E}^\sigma_{\mathbf{z}}(f_\rho))-(\mathcal{E}^\sigma_{\mathbf{z}}(f_j)-\mathcal{E}^\sigma_{\mathbf{z}}(f_\rho))|}{\sqrt{\mathcal{E}^\sigma(f)-\mathcal{E}^\sigma(f_\rho)+2\varepsilon}}\le\sqrt{\varepsilon}\tag{12}$$

and

$$\frac{|(\mathcal{E}^\sigma(f)-\mathcal{E}^\sigma(f_\rho))-(\mathcal{E}^\sigma(f_j)-\mathcal{E}^\sigma(f_\rho))|}{\sqrt{\mathcal{E}^\sigma(f)-\mathcal{E}^\sigma(f_\rho)+2\varepsilon}}\le\sqrt{\varepsilon}.\tag{13}$$

The latter inequality together with the fact that $\varepsilon\le\mathcal{E}^\sigma(f)-\mathcal{E}^\sigma(f_\rho)+2\varepsilon$ implies

$$\begin{aligned}
\mathcal{E}^\sigma(f_j)-\mathcal{E}^\sigma(f_\rho)+2\varepsilon&=(\mathcal{E}^\sigma(f_j)-\mathcal{E}^\sigma(f_\rho))-(\mathcal{E}^\sigma(f)-\mathcal{E}^\sigma(f_\rho))+\mathcal{E}^\sigma(f)-\mathcal{E}^\sigma(f_\rho)+2\varepsilon\\
&\le\sqrt{\varepsilon}\sqrt{(\mathcal{E}^\sigma(f)-\mathcal{E}^\sigma(f_\rho))+2\varepsilon}+\mathcal{E}^\sigma(f)-\mathcal{E}^\sigma(f_\rho)+2\varepsilon\\
&\le 2(\mathcal{E}^\sigma(f)-\mathcal{E}^\sigma(f_\rho)+2\varepsilon).
\end{aligned}\tag{14}$$

For any $f\in\mathcal{H}$, if the following inequality holds

$$\frac{(\mathcal{E}^\sigma(f)-\mathcal{E}^\sigma(f_\rho))-(\mathcal{E}^\sigma_{\mathbf{z}}(f)-\mathcal{E}^\sigma_{\mathbf{z}}(f_\rho))}{\sqrt{\mathcal{E}^\sigma(f)-\mathcal{E}^\sigma(f_\rho)+2\varepsilon}}>4\sqrt{\varepsilon},$$

then combining estimates in (12) and (13) we know that there holds

$$\frac{(\mathcal{E}^\sigma(f_j)-\mathcal{E}^\sigma(f_\rho))-(\mathcal{E}^\sigma_{\mathbf{z}}(f_j)-\mathcal{E}^\sigma_{\mathbf{z}}(f_\rho))}{\sqrt{\mathcal{E}^\sigma(f)-\mathcal{E}^\sigma(f_\rho)+2\varepsilon}}>2\sqrt{\varepsilon}.$$

This together with inequality (14) tells us that the following inequality holds

$$\frac{(\mathcal{E}^\sigma(f_j)-\mathcal{E}^\sigma(f_\rho))-(\mathcal{E}^\sigma_{\mathbf{z}}(f_j)-\mathcal{E}^\sigma_{\mathbf{z}}(f_\rho))}{\sqrt{\mathcal{E}^\sigma(f_j)-\mathcal{E}^\sigma(f_\rho)+2\varepsilon}}>\sqrt{\varepsilon}.$$

Consequently, based on the above estimates, we come to the following conclusion

$$\begin{aligned}&\mathrm{Prob}_{\mathbf{z}\in\mathcal{Z}^m}\left\{\sup_{f\in\mathcal{H}}\frac{(\mathcal{E}^\sigma(f)-\mathcal{E}^\sigma(f_\rho))-(\mathcal{E}^\sigma_{\mathbf{z}}(f)-\mathcal{E}^\sigma_{\mathbf{z}}(f_\rho))}{\sqrt{\mathcal{E}^\sigma(f)-\mathcal{E}^\sigma(f_\rho)+2\varepsilon}}>4\sqrt{\varepsilon}\right\}\\ &\le\sum_{j=1}^{J}\mathrm{Prob}_{\mathbf{z}\in\mathcal{Z}^m}\left\{\frac{(\mathcal{E}^\sigma(f_j)-\mathcal{E}^\sigma(f_\rho))-(\mathcal{E}^\sigma_{\mathbf{z}}(f_j)-\mathcal{E}^\sigma_{\mathbf{z}}(f_\rho))}{\sqrt{\mathcal{E}^\sigma(f_j)-\mathcal{E}^\sigma(f_\rho)+2\varepsilon}}>\sqrt{\varepsilon}\right\}\\ &\le\mathcal{N}\left(\mathcal{H},\frac{\varepsilon}{\sqrt{2/e}\sigma}\right)\exp\left\{-\frac{3m\varepsilon}{4\sqrt{2/e}\sup_{f\in\mathcal{H}}\|f-f_\rho\|_\infty\sigma+6c_{\mathcal{H},\rho,0}}\right\}.\end{aligned}$$

This completes the proof of Proposition 15. ■

Proof [Proof of Proposition 11] From the Complexity Assumption I, we know that

$$\mathcal{N}\left(\mathcal{H},\varepsilon/(\sqrt{2/e}\sigma)\right)\le\exp\left\{c_{I,p}(\sqrt{2/e})^p\sigma^p/\varepsilon^p\right\}.$$

This in connection with Proposition 15 yields

$$\begin{aligned}&\mathrm{Prob}_{\mathbf{z}\in\mathcal{Z}^m}\left\{\sup_{f\in\mathcal{H}}\frac{(\mathcal{E}^\sigma(f)-\mathcal{E}^\sigma(f_\rho))-(\mathcal{E}^\sigma_{\mathbf{z}}(f)-\mathcal{E}^\sigma_{\mathbf{z}}(f_\rho))}{\sqrt{\mathcal{E}^\sigma(f)-\mathcal{E}^\sigma(f_\rho)+2\varepsilon}}>4\sqrt{\varepsilon}\right\}\\ &\le\exp\left\{\frac{A_p\sigma^p}{\varepsilon^p}-\frac{m\varepsilon}{\sigma B_{\mathcal{H},\rho}+2c_{\mathcal{H},\rho,0}}\right\},\end{aligned}$$

where A_p and $B_{\mathcal{H},\rho}$ are positive constants given by

$$A_p=c_{I,p}(\sqrt{2/e})^p\ \text{and}\ \ B_{\mathcal{H},\rho}=4\sqrt{2/e}\sup_{f\in\mathcal{H}}\|f-f_\rho\|_\infty/3.$$

By setting

$$\exp\left\{\frac{A_p\sigma^p}{\varepsilon^p}-\frac{m\varepsilon}{\sigma B_{\mathcal{H},\rho}+2c_{\mathcal{H},\rho,0}}\right\}\le\frac{\delta}{2},$$

we obtain

$$\varepsilon^{p+1}-\frac{\log(2/\delta)\left(\sigma B_{\mathcal{H},\rho}+2c_{\mathcal{H},\rho,0}\right)}{m}\varepsilon^p-\frac{A_p\left(\sigma B_{\mathcal{H},\rho}+2c_{\mathcal{H},\rho,0}\right)\sigma^p}{m}\ge 0.$$

Lemma 7.2 in Cucker and Zhou (2007) tells us that the above inequality holds if

$$\varepsilon\ge\max\left\{\frac{c_{\mathcal{H},\rho}}{\sigma^2},\frac{2\log(2/\delta)\left(\sigma B_{\mathcal{H},\rho}+2c_{\mathcal{H},\rho,0}\right)}{m},\left(\frac{2A_p\left(\sigma B_{\mathcal{H},\rho}+2c_{\mathcal{H},\rho,0}\right)\sigma^p}{m}\right)^{1/(1+p)}\right\}.$$

In view of the above condition, we choose a sufficient large $\varepsilon_{\mathcal{H},\rho}$ as follows

$$\varepsilon_{\mathcal{H},\rho} = c_{\mathcal{H},\rho,1} \log(2/\delta)(\sigma^{-2} + \sigma m^{-1/(1+p)}),$$

where $c_{\mathcal{H},\rho,1}$ is a positive constant independent of m, σ or δ and given by

$$c_{\mathcal{H},\rho,1} = 2c_{\mathcal{H},\rho} + 2(A_p + 1)(B_{\mathcal{H},\rho} + 2c_{\mathcal{H},\rho,0}).$$

With the above choice of $\varepsilon_{\mathcal{H},\rho}$ and following the above discussions, we see that for any $0 < \delta < 1$, with confidence $1 - \delta/2$, there holds

$$\sup_{f\in\mathcal{H}} \left\{ ((\mathcal{E}^\sigma(f) - \mathcal{E}^\sigma(f_\rho)) - (\mathcal{E}^\sigma_{\mathbf{z}}(f) - \mathcal{E}^\sigma_{\mathbf{z}}(f_\rho))) \Big/ \sqrt{\mathcal{E}^\sigma(f) - \mathcal{E}^\sigma(f_\rho) + \varepsilon_{\mathcal{H},\rho}} \right\} \leq 4\sqrt{\varepsilon_{\mathcal{H},\rho}},$$

which yields

$$(\mathcal{E}^\sigma(f_{\mathbf{z}}) - \mathcal{E}^\sigma(f_\rho)) - (\mathcal{E}^\sigma_{\mathbf{z}}(f_{\mathbf{z}}) - \mathcal{E}^\sigma_{\mathbf{z}}(f_\rho)) \leq 4\sqrt{\varepsilon_{\mathcal{H},\rho}} \sqrt{\mathcal{E}^\sigma(f_{\mathbf{z}}) - \mathcal{E}^\sigma(f_\rho) + 2\varepsilon_{\mathcal{H},\rho}}.$$

Applying the basic inequality $\sqrt{ab} \leq (a+b)/2$ for $a, b \geq 0$, we know that for any $0 < \delta < 1$, with confidence $1 - \delta$, there holds[2]

$$\mathcal{S}_2(\mathbf{z}) = (\mathcal{E}^\sigma(f_{\mathbf{z}}) - \mathcal{E}^\sigma(f_\rho)) - (\mathcal{E}^\sigma_{\mathbf{z}}(f_{\mathbf{z}}) - \mathcal{E}^\sigma_{\mathbf{z}}(f_\rho)) \leq \frac{1}{2}(\mathcal{E}^\sigma(f_{\mathbf{z}}) - \mathcal{E}^\sigma(f_\rho)) + 9\varepsilon_{\mathcal{H},\rho}. \tag{15}$$

Proposition 9 tells us that

$$\begin{aligned} \mathcal{E}^\sigma(f_{\mathbf{z}}) - \mathcal{E}^\sigma(f_\rho) &= \mathcal{E}^\sigma(f_{\mathbf{z}}) - \mathcal{E}^\sigma(f^\sigma_{\mathcal{H}}) + \mathcal{E}^\sigma(f^\sigma_{\mathcal{H}}) - \mathcal{E}^\sigma(f_\rho) \\ &\leq \mathcal{S}_1(\mathbf{z}) + \mathcal{S}_2(\mathbf{z}) + \|f_{\mathcal{H}} - f_\rho\|^2_{\mathcal{L}^2_{\rho_{\mathcal{X}}}} + c_{\mathcal{H},\rho}/\sigma^2, \end{aligned} \tag{16}$$

where the above inequality is due to Lemma 7 and the observation that

$$\begin{aligned} \mathcal{E}^\sigma(f^\sigma_{\mathcal{H}}) - \mathcal{E}^\sigma(f_\rho) &= \mathcal{E}^\sigma(f^\sigma_{\mathcal{H}}) - \mathcal{E}^\sigma(f_{\mathcal{H}}) + \mathcal{E}^\sigma(f_{\mathcal{H}}) - \mathcal{E}^\sigma(f_\rho) \\ &\leq \mathcal{E}^\sigma(f_{\mathcal{H}}) - \mathcal{E}^\sigma(f_\rho) \\ &\leq \|f_{\mathcal{H}} - f_\rho\|^2_{\mathcal{L}^2_{\rho_{\mathcal{X}}}} + c_{\mathcal{H},\rho}/\sigma^2. \end{aligned}$$

Combining estimates in (15) and (16), we come to the conclusion that for any $0 < \delta < 1$, with confidence $1 - \delta/2$, there holds

$$\mathcal{S}_2(\mathbf{z}) \leq \frac{1}{2}(\mathcal{S}_1(\mathbf{z}) + \mathcal{S}_2(\mathbf{z})) + \frac{1}{2}\|f_{\mathcal{H}} - f_\rho\|^2_{\mathcal{L}^2_{\rho_{\mathcal{X}}}} + C_{\mathcal{H},\rho,2}\left(\log\frac{2}{\delta}\right)\left\{\frac{1}{\sigma^2} + \frac{\sigma}{m^{1/(1+p)}}\right\},$$

where $C_{\mathcal{H},\rho,2}$ is a positive constant independent of m, σ or δ and given by $C_{\mathcal{H},\rho,2} = 2c_{\mathcal{H},\rho} + 9c_{\mathcal{H},\rho,1}$. This completes the proof of Proposition 11. ■

2. Similarly, refined estimate can be also derived here by using Young's inequality $ab \leq \frac{ta^2}{2} + \frac{b^2}{2t}$ for $a, b \in \mathbb{R}$, $t > 0$. In our proof, again we choose $t = 1$ for simplification.

4.3.4 Proof of Theorem 4

Proof From Lemma 7 and Proposition 9, we know that

$$\|f_{\mathbf{z}} - f_\rho\|^2_{\mathcal{L}^2_{\rho_\mathcal{X}}} \leq \mathcal{S}_1(\mathbf{z}) + \mathcal{S}_2(\mathbf{z}) + \|f_\mathcal{H} - f_\rho\|^2_{\mathcal{L}^2_{\rho_\mathcal{X}}} + 2c_{\mathcal{H},\rho}/\sigma^2. \tag{17}$$

Combining estimates in Proposition 10 and Proposition 11 for the sample error terms $\mathcal{S}_1(\mathbf{z})$ and $\mathcal{S}_2(\mathbf{z})$, we know that for any $0 < \delta < 1$, with confidence $1 - \delta$, there holds

$$\mathcal{S}_1(\mathbf{z}) + \mathcal{S}_2(\mathbf{z}) \leq 2\|f_\mathcal{H} - f_\rho\|^2_{\mathcal{L}^2_{\rho_\mathcal{X}}} + (2C_{\mathcal{H},\rho,1} + 4C_{\mathcal{H},\rho,2}) \log(2/\delta) \{\sigma^{-2} + \sigma m^{-1/(1+p)}\}.$$

This in connection with the estimate in (17) tells us that for any $0 < \delta < 1$, with confidence $1 - \delta$, there holds

$$\|f_{\mathbf{z}} - f_\rho\|^2_{\mathcal{L}^2_{\rho_\mathcal{X}}} \leq 3\|f_\mathcal{H} - f_\rho\|^2_{\mathcal{L}^2_{\rho_\mathcal{X}}} + C_{\mathcal{H},\rho} \log(2/\delta) \{\sigma^{-2} + \sigma m^{-1/(1+p)}\},$$

where $C_{\mathcal{H},\rho} = 2C_{\mathcal{H},\rho,1} + 4C_{\mathcal{H},\rho,2} + 4c_{\mathcal{H},\rho}$. This completes the proof of Theorem 4. ■

4.3.5 Proof of Theorem 5

The proof of Theorem 5 can be similarly conducted as that of Theorem 4, since the error decomposition in Proposition 9 holds when Y is bounded. Therefore, we also need to bound the two sample error terms $\mathcal{S}_1(\mathbf{z})$ and $\mathcal{S}_1(\mathbf{z})$, respectively.

Proposition 16 *Assume that $|y| \leq M$ almost surely for some $M > 0$, and $f_\rho \in \mathcal{H}$. For any $0 < \delta < 1$, with confidence $1 - \delta/2$, there holds*

$$\mathcal{S}_1(\mathbf{z}) \leq C'_{\mathcal{H},\rho,1} \log(2/\delta)(\sigma^{-2} + m^{-1}),$$

where $C'_{\mathcal{H},\rho,1}$ is a positive constant that independent of m, σ or δ and will be given explicitly in the proof.

Proof We will finish the proof by following similar process as done for Proposition 10. We first introduce the random variable $\bar{\xi}_1(z)$ as follows

$$\bar{\xi}_1(z) = -\sigma^2 \exp\left\{-(y - f^\sigma_\mathcal{H}(x))^2/\sigma^2\right\} + \sigma^2 \exp\left\{-(y - f_\rho(x))^2/\sigma^2\right\}, \ z \in \mathcal{Z}.$$

It follows from the proof of Proposition 10 and the boundedness of Y that for any $z \in \mathcal{Z}$, there holds

$$\begin{aligned} |\bar{\xi}_1(z)| &\leq \left|(2y - f^\sigma_\mathcal{H}(x) - f_\rho(x))(f^\sigma_\mathcal{H}(x) - f_\rho(x))\right| \\ &\leq \Big(2M + \|f_\rho\|_\infty + \sup_{f\in\mathcal{H}} \|f\|_\infty\Big) \sup_{f\in\mathcal{H}} \|f - f_\rho\|_\infty. \end{aligned}$$

Consequently, the following estimate holds

$$\left|\bar{\xi}_1 - \mathbb{E}\bar{\xi}_1\right| \leq 2\Big(2M + \|f_\rho\|_\infty + \sup_{f\in\mathcal{H}} \|f\|_\infty\Big) \sup_{f\in\mathcal{H}} \|f - f_\rho\|_\infty := c'_{\mathcal{H},\rho,0}.$$

Denote the variance of the random variable $\bar{\xi}_1$ as $\mathrm{var}(\bar{\xi}_1)$. From the proof of Proposition 10 and the boundedness of Y, we have

$$\begin{aligned}\mathrm{var}(\bar{\xi}_1) &= \mathbb{E}\bar{\xi}_1^2 - (\mathbb{E}\bar{\xi}_1)^2 \\ &\leq \mathbb{E}\bar{\xi}_1^2 \leq \mathbb{E}\left((f_{\mathcal{H}}^{\sigma}(x) - f_\rho(x))^2(2y - f_{\mathcal{H}}^{\sigma}(x) - f_\rho(x))^2\right) \\ &\leq \left(12M^2 + 3\sup_{f\in\mathcal{H}}\|f\|_\infty^2 + 3\|f_\rho\|_\infty^2\right)\int_{\mathcal{X}}(f_{\mathcal{H}}^{\sigma}(x) - f_\rho(x))^2 d\rho_{\mathcal{X}}(x).\end{aligned}$$

Recalling the fact that $f_\rho \in \mathcal{H}$, as a consequence of Lemma 7, we obtain

$$\int_{\mathcal{X}}(f_{\mathcal{H}}^{\sigma}(x) - f_\rho(x))^2 d\rho_{\mathcal{X}}(x) \leq \int_{\mathcal{X}}(f_{\mathcal{H}}(x) - f_\rho(x))^2 d\rho_{\mathcal{X}}(x) + \frac{2c_{\mathcal{H},\rho}}{\sigma^2} = \frac{2c_{\mathcal{H},\rho}}{\sigma^2}.$$

Combining the above two estimates, we obtain the following upper bound for the variance of $\bar{\xi}_1$:

$$\mathrm{var}(\bar{\xi}_1) \leq c'_{\mathcal{H},\rho,1}/\sigma^2 \quad \text{with} \quad c'_{\mathcal{H},\rho,1} = 2c_{\mathcal{H},\rho}\Big(12M^2 + 3\sup_{f\in\mathcal{H}}\|f\|_\infty^2 + 3\|f_\rho\|_\infty^2\Big).$$

Applying the one-sided Bernstein's inequality in Lemma 13 to the random variable $\bar{\xi}_1$ and with simple computations, we come to the conclusion that for any $0 < \delta < 1$, with confidence $1 - \delta/2$, there holds

$$\mathcal{S}_1(\mathbf{z}) \leq C'_{\mathcal{H},\rho,1}\log(2/\delta)(\sigma^{-2} + m^{-1}),$$

where $C'_{\mathcal{H},\rho,1}$ is a positive constant independent of m, σ or δ and given by $C'_{\mathcal{H},\rho,1} = 2 + c'_{\mathcal{H},\rho,1}/2 + 2c'_{\mathcal{H},\rho,0}/3$. This completes the proof. ■

We now turn to bound the sample error term $\mathcal{S}_2(\mathbf{z})$ when Y is bounded.

Proposition 17 *Assume that the Complexity Assumption II with $0 < s < 2$ holds, $|y| \leq M$ almost surely for some $M > 0$. Let $f_\rho \in \mathcal{H}$ and $\sigma \geq 1$. For any $f \in \mathcal{H}$ and $0 < \delta < 1$, with confidence $1 - \delta/2$, there holds*

$$\{\mathcal{E}^\sigma(f) - \mathcal{E}^\sigma(f_\rho)\} - \{\mathcal{E}_{\mathbf{z}}^\sigma(f) - \mathcal{E}_{\mathbf{z}}^\sigma(f_\rho)\} \leq \frac{1}{2}\{\mathcal{E}^\sigma(f) - \mathcal{E}^\sigma(f_\rho)\} + C'_{\mathcal{H},\rho,2}\log(2/\delta)m^{-\frac{2}{2+s}},$$

where $C'_{\mathcal{H},\rho,2}$ is a positive constant independent of m, σ or δ and will be given explicitly in the proof.

Proof To prove the proposition, we apply Lemma 14 to the function set $\mathcal{F}_{\mathcal{H}}$, which is defined as

$$\mathcal{F}_{\mathcal{H}} = \left\{g \;\middle|\; g(z) = \ell_\sigma(y, f(x)) - \ell_\sigma(y, f_\rho(x)) + \frac{c_{\mathcal{H},\rho}}{\sigma^2}, f \in \mathcal{H}, z \in \mathcal{Z}\right\}.$$

According to the definition of $\mathcal{F}_{\mathcal{H}}$, for any $g \in \mathcal{F}_{\mathcal{H}}$, it can be explicitly expressed as

$$g(z) = -\sigma^2\exp\left\{-(y - f(x))^2/\sigma^2\right\} + \sigma^2\exp\left\{-(y - f_\rho(x))^2/\sigma^2\right\} + \frac{c_{\mathcal{H},\rho}}{\sigma^2},$$

with $z \in \mathcal{Z}$ and $f \in \mathcal{H}$. Recalling that $|y| \le M$ almost surely and $\sigma \ge 1$, simple computations show that

$$\|g\|_\infty \le \Big(2M + \|f_\rho\|_\infty + \sup_{f\in\mathcal{H}} \|f\|_\infty\Big) \sup_{f\in\mathcal{H}} \|f - f_\rho\|_\infty + c_{\mathcal{H},\rho}.$$

Applying the mean value theorem again as done in the proof of Proposition 10, we get

$$\begin{aligned}
&\left(-\sigma^2 \exp\left\{-(y-f(x))^2/\sigma^2\right\} + \sigma^2 \exp\left\{-(y-f_\rho(x))^2/\sigma^2\right\}\right)^2 \\
&\qquad \le \left((y-f(x))^2 - (y-f_\rho(x))^2\right)^2 \\
&\qquad \le \Big(2M + \sup_{f\in\mathcal{H}} \|f\|_\infty + \|f_\rho\|_\infty\Big)^2 (f(x) - f_\rho(x))^2,
\end{aligned}$$

where the last inequality is again due to the boundedness of Y. This in connection with Lemma 7 tells us that

$$\begin{aligned}
\mathbb{E}g^2 &= \int_{\mathcal{Z}} \left(-\sigma^2 \exp\left\{-\frac{(y-f(x))^2}{\sigma^2}\right\} + \sigma^2 \exp\left\{-\frac{(y-f_\rho(x))^2}{\sigma^2}\right\}\right)^2 d\rho \\
&\quad + \frac{2c_{\mathcal{H},\rho}}{\sigma^2} \int_{\mathcal{Z}} \left(-\sigma^2 \exp\left\{-\frac{(y-f(x))^2}{\sigma^2}\right\} + \sigma^2 \exp\left\{-\frac{(y-f_\rho(x))^2}{\sigma^2}\right\}\right) d\rho + \frac{c_{\mathcal{H},\rho}^2}{\sigma^4} \\
&\le \Big(2M + \sup_{f\in\mathcal{H}} \|f\|_\infty + \|f_\rho\|_\infty\Big)^2 (\mathcal{E}(f) - \mathcal{E}(f_\rho)) + \frac{2c_{\mathcal{H},\rho}}{\sigma^2} \left(\mathcal{E}^\sigma(f) - \mathcal{E}^\sigma(f_\rho) + \frac{c_{\mathcal{H},\rho}}{\sigma^2}\right) \\
&\le \Big(2M + \sup_{f\in\mathcal{H}} \|f\|_\infty + \|f_\rho\|_\infty\Big)^2 \left(\mathcal{E}^\sigma(f) - \mathcal{E}^\sigma(f_\rho) + \frac{c_{\mathcal{H},\rho}}{\sigma^2}\right) + \frac{2c_{\mathcal{H},\rho}}{\sigma^2} \left(\mathcal{E}^\sigma(f) - \mathcal{E}^\sigma(f_\rho) + \frac{c_{\mathcal{H},\rho}}{\sigma^2}\right) \\
&= \left(\Big(2M + \sup_{f\in\mathcal{H}} \|f\|_\infty + \|f_\rho\|_\infty\Big)^2 + \frac{2c_{\mathcal{H},\rho}}{\sigma^2}\right) \left(\mathcal{E}^\sigma(f) - \mathcal{E}^\sigma(f_\rho) + \frac{c_{\mathcal{H},\rho}}{\sigma^2}\right) \\
&= \left(\Big(2M + \sup_{f\in\mathcal{H}} \|f\|_\infty + \|f_\rho\|_\infty\Big)^2 + \frac{2c_{\mathcal{H},\rho}}{\sigma^2}\right) \mathbb{E}g \\
&\le \left(\Big(2M + \sup_{f\in\mathcal{H}} \|f\|_\infty + \|f_\rho\|_\infty\Big)^2 + 2c_{\mathcal{H},\rho}\right) \mathbb{E}g,
\end{aligned}$$

where the last inequality is due to the assumption that $\sigma \ge 1$.

For any g_1, $g_2 \in \mathcal{F}_\mathcal{H}$, there exist f_1, $f_2 \in \mathcal{H}$ such that

$$g_1(z) = -\sigma^2 \exp\left\{-(y-f_1(x))^2/\sigma^2\right\} + \sigma^2 \exp\left\{-(y-f_\rho(x))^2/\sigma^2\right\} + \frac{c_{\mathcal{H},\rho}}{\sigma^2}$$

and

$$g_2(z) = -\sigma^2 \exp\left\{-(y-f_2(x))^2/\sigma^2\right\} + \sigma^2 \exp\left\{-(y-f_\rho(x))^2/\sigma^2\right\} + \frac{c_{\mathcal{H},\rho}}{\sigma^2}.$$

Applying the mean value theorem and noticing the boundedness of Y, we have

$$|g_1(z) - g_2(z)| \le 2\Big(M + \sup_{f\in\mathcal{H}} \|f\|_\infty\Big)\|f_1 - f_2\|_\infty,\ z \in \mathcal{Z}.$$

Under the Complexity Assumption II with $0 < s < 2$, the following relation between the ℓ^2-empirical covering numbers of $\mathcal{F}_{\mathcal{H}}$ and $\mathcal{H}$ holds

$$\log \mathcal{N}_2(\mathcal{F}_{\mathcal{H}}, \eta) \leq \log \mathcal{N}_2\Big(\mathcal{H}, \eta \Big/ \Big(2M + 2\sup_{f\in\mathcal{H}} \|f\|_\infty\Big)\Big) \leq c_{II,s}\Big(\Big(2M + 2\sup_{f\in\mathcal{H}} \|f\|_\infty\Big)\Big/\eta\Big)^s.$$

For notation simplification, we denote

$$\begin{aligned}
c'_{\mathcal{H},\rho,2} &= \Big(2M + \sup_{f\in\mathcal{H}} \|f\|_\infty + \|f_\rho\|_\infty\Big)^2 + 2c_{\mathcal{H},\rho},\\
B'_{\mathcal{H},\rho} &= \Big(2M + \sup_{f\in\mathcal{H}} \|f\|_\infty + \|f_\rho\|_\infty\Big) \sup_{f\in\mathcal{H}} \|f - f_\rho\|_\infty + c_{\mathcal{H},\rho},\\
a_{\mathcal{H},s} &= c_{II,s}\Big(2M + 2\sup_{f\in\mathcal{H}} \|f\|_\infty\Big)^s.
\end{aligned}$$

Applying Lemma 14 to the function set $\mathcal{F}_{\mathcal{H}}$, with simple computations, we come to the conclusion that when $\sigma \geq 1$, for any $0 < \delta < 1$ with confidence $1 - \delta/2$, there holds

$$\{\mathcal{E}^\sigma(f) - \mathcal{E}^\sigma(f_\rho)\} - \{\mathcal{E}^\sigma_{\mathbf{z}}(f) - \mathcal{E}^\sigma_{\mathbf{z}}(f_\rho)\} \leq \frac{1}{2}\{\mathcal{E}^\sigma(f) - \mathcal{E}^\sigma(f_\rho)\} + C'_{\mathcal{H},\rho,2}\log(2/\delta)m^{-\frac{2}{2+s}},$$

where $C'_{\mathcal{H},\rho,2}$ is a positive constant independent of m, σ or δ and given by

$$C'_{\mathcal{H},\rho,2} = 18B'_{\mathcal{H},\rho} + 2c'_{\mathcal{H},\rho,2} + 2a_s a^{2/(2+s)}_{\mathcal{H},s}(c'_{\mathcal{H},\rho,2} + B'_{\mathcal{H},\rho})^{(2-s)/(2+s)},$$

and a_s is a positive constant depending only on s. This completes the proof of Proposition 17. ■

Proof [Proof of Theorem 5] Following from the estimate in inequality (11), and recalling that $f_\rho \in \mathcal{H}$, we have

$$\|f_{\mathbf{z}} - f_\rho\|^2_{\mathcal{L}^2_{\rho_{\mathcal{X}}}} \leq \mathcal{S}_1(\mathbf{z}) + \mathcal{S}_2(\mathbf{z}) + 2c_{\mathcal{H},\rho}/\sigma^2. \tag{18}$$

As a consequence of Proposition 17, we know that when $\sigma \geq 1$, for any $0 < \delta < 1$ with confidence $1 - \delta/2$, there holds

$$\{\mathcal{E}^\sigma(f_{\mathbf{z}}) - \mathcal{E}^\sigma(f_\rho)\} - \{\mathcal{E}^\sigma_{\mathbf{z}}(f_{\mathbf{z}}) - \mathcal{E}^\sigma_{\mathbf{z}}(f_\rho)\} \leq \frac{1}{2}\{\mathcal{E}^\sigma(f_{\mathbf{z}}) - \mathcal{E}^\sigma(f_\rho)\} + C'_{\mathcal{H},\rho,2}\log(2/\delta)m^{-\frac{2}{2+s}}.$$

The above inequality together with Lemma 7 yields

$$\begin{aligned}
\mathcal{S}_2(\mathbf{z}) &= \{\mathcal{E}^\sigma(f_{\mathbf{z}}) - \mathcal{E}^\sigma(f_\rho)\} - \{\mathcal{E}^\sigma_{\mathbf{z}}(f_{\mathbf{z}}) - \mathcal{E}^\sigma_{\mathbf{z}}(f_\rho)\}\\
&\leq \frac{1}{2}\|f_{\mathbf{z}} - f_\rho\|^2_{\mathcal{L}^2_{\rho_{\mathcal{X}}}} + \frac{c_{\mathcal{H},\rho}}{2\sigma^2} + C'_{\mathcal{H},\rho,2}\log(2/\delta)m^{-\frac{2}{2+s}}.
\end{aligned}$$

This in connection with the upper bound for the sample error term $\mathcal{S}_1(\mathbf{z})$ in Proposition 16 and inequality (18), with the choice $\sigma = m^{1/(2+s)}$, yields that for any $0 < \delta < 1$, with confidence $1 - \delta$, there holds

$$\|f_{\mathbf{z}} - f_\rho\|^2_{\mathcal{L}^2_{\rho_{\mathcal{X}}}} \leq C'_{\mathcal{H},\rho}\log(2/\delta)m^{-\frac{2}{2+s}},$$

where $C'_{\mathcal{H},\rho} = 2C'_{\mathcal{H},\rho,1} + C'_{\mathcal{H},\rho,2} + 3c_{\mathcal{H},\rho}$. This completes the proof of Theorem 5. ■

4.3.6 Proof of Theorem 6

To prove Theorem 6, we first prove the following conclusion.

Lemma 18 *Assume that the Noise Assumption holds, and $f_\rho \in \mathcal{H}$. Let σ be fixed and satisfy*

$$\sigma > \sigma_{\mathcal{H},\rho} = \sqrt{2}\Big(M_0 + \|f_\rho\|_\infty + \sup_{f\in\mathcal{H}}\|f\|_\infty\Big).$$

For any $f \in \mathcal{H}$, there exists a positive constant $c_{\mathcal{H},\sigma,\rho} \in (0,1)$, such that

$$c_{\mathcal{H},\sigma,\rho}\left\{\mathcal{E}(f) - \mathcal{E}(f_\rho)\right\} \le \mathcal{E}^\sigma(f) - \mathcal{E}^\sigma(f_\rho).$$

Proof Under the Noise Assumption, when $\sigma > \sigma_{\mathcal{H},\rho}$, Theorem 8 shows that for any $f \in \mathcal{H}$,

$$\mathcal{E}^\sigma(f) - \mathcal{E}^\sigma(f_{\mathcal{H}}^\sigma) = \mathcal{E}^\sigma(f) - \mathcal{E}^\sigma(f_{\mathcal{H}}) = \mathcal{E}^\sigma(f) - \mathcal{E}^\sigma(f_\rho).$$

For any $x \in \mathcal{X}$, again we denote $F_x(u) = 1 - \int_{-M_0}^{M_0} \exp\left\{-\frac{(t-u)^2}{\sigma^2}\right\} p_{\epsilon|X=x}(t)dt$, then

$$\begin{aligned}
\mathcal{E}^\sigma(f) - \mathcal{E}^\sigma(f_\rho) &= \sigma^2 \int_{\mathcal{X}} \left(F_x(f(x) - f_\rho(x)) - F_x(0)\right) d\rho_{\mathcal{X}}(x) \\
&= \sigma^2 \int_{\mathcal{X}} \left\{F_x'(0)(f(x) - f_\rho(x)) + \frac{F_x''(\xi_x)}{2}(f(x) - f_\rho(x))^2\right\} d\rho_{\mathcal{X}}(x) \\
&= \int_{\mathcal{X}} \frac{\sigma^2 F_x''(\xi_x)}{2}(f(x) - f_\rho(x))^2 d\rho_{\mathcal{X}}(x),
\end{aligned}$$

where the last equality follows from the fact that $F_x'(0) = 0$ and ξ_x falls between 0 and $f(x) - f_\rho(x)$ for any $x \in \mathcal{X}$. It is easy to see that when σ is fixed and $\sigma > \sigma_{\mathcal{H},\rho}$, we have

$$\begin{aligned}
F_x''(\xi_x) &= 2\int_{-M_0}^{M_0} \exp\left\{-\frac{(t-\xi_x)^2}{\sigma^2}\right\}\left(\frac{\sigma^2 - 2(t-\xi_x)^2}{\sigma^4}\right) p_{\epsilon|X=x}(t)dt \\
&\ge (2\sigma^2 - 2\sigma_{\mathcal{H},\rho}^2)/\sigma^4 \exp(-\sigma_{\mathcal{H},\rho}^2/\sigma^2), \text{ for any } x \in \mathcal{X},
\end{aligned}$$

where the last inequality is due to the following fact

$$|t - \xi_x| \le \sqrt{2}\sigma_{\mathcal{H},\rho}/2, \quad t \in [-M_0, M_0],\, x \in \mathcal{X}.$$

As a result, we come to the conclusion that

$$\mathcal{E}^\sigma(f) - \mathcal{E}^\sigma(f_\rho) \ge c_{\mathcal{H},\sigma,\rho}\left\{\mathcal{E}(f) - \mathcal{E}(f_\rho)\right\},$$

where $c_{\mathcal{H},\sigma,\rho} = (\sigma^2 - \sigma_{\mathcal{H},\rho}^2)/\sigma^2 \exp(-\sigma_{\mathcal{H},\rho}^2/\sigma^2)$. Noticing that $0 < c_{\mathcal{H},\sigma,\rho} < 1$, we have verified our assertion. ■

The proof of Theorem 6 is different from the proofs of Theorem 4 and Theorem 5. This is because when σ is fixed, σ^{-1} does not tend to zero and consequently we cannot get

meaningful convergence rates via the error decomposition in Proposition 9. However, from Lemma 18, we know that

$$\|f_{\mathbf{z}} - f_\rho\|^2_{\mathcal{L}^2_{\rho_\mathcal{X}}} \leq c^{-1}_{\mathcal{H},\sigma,\rho} \left\{\mathcal{E}^\sigma(f_{\mathbf{z}}) - \mathcal{E}^\sigma(f_\rho)\right\} = c^{-1}_{\mathcal{H},\sigma,\rho} \left(\mathcal{S}_1(\mathbf{z}) + \mathcal{S}_2(\mathbf{z})\right),$$

where the definitions of $\mathcal{S}_1(\mathbf{z})$ and $\mathcal{S}_2(\mathbf{z})$ are inherited from Proposition 9.

We notice that under the condition that the Noise Assumption holds, and $f_\rho \in \mathcal{H}$, when σ is fixed and satisfies

$$\sigma > \sigma_{\mathcal{H},\rho} = \sqrt{2}\Big(M_0 + \|f_\rho\|_\infty + \sup_{f\in\mathcal{H}} \|f\|_\infty\Big),$$

Theorem 8 tells us that almost surely $f^\sigma_\mathcal{H} = f_\rho$. In this situation, almost surely we have $\mathcal{S}_1(\mathbf{z}) = 0$. Therefore, to prove Theorem 6, it suffices to bound the sample error term $\mathcal{S}_2(\mathbf{z})$. This can be done by applying Lemma 14 to the function set

$$\mathcal{F}_\mathcal{H} = \left\{g \,\middle|\, g(z) = \ell_\sigma(y, f(x)) - \ell_\sigma(y, f_\rho(x)) : f \in \mathcal{H}, z \in \mathcal{Z}\right\}.$$

Proposition 19 *Assume that the Complexity Assumption II with $0 < s < 2$ and the Noise Assumption hold. Let $f_\rho \in \mathcal{H}$, σ be fixed and satisfy*

$$\sigma > \sigma_{\mathcal{H},\rho} = \sqrt{2}\Big(M_0 + \|f_\rho\|_\infty + \sup_{f\in\mathcal{H}} \|f\|_\infty\Big).$$

For any $f \in \mathcal{H}$ and $0 < \delta < 1$, with confidence $1 - \delta$, there holds

$$\left\{\mathcal{E}^\sigma(f) - \mathcal{E}^\sigma(f_\rho)\right\} - \left\{\mathcal{E}^\sigma_{\mathbf{z}}(f) - \mathcal{E}^\sigma_{\mathbf{z}}(f_\rho)\right\} \leq \frac{1}{2}\left\{\mathcal{E}^\sigma(f) - \mathcal{E}^\sigma(f_\rho)\right\} + C_{\mathcal{H},\sigma,\rho,1} \log(1/\delta) m^{-\frac{2}{2+s}},$$

where $C_{\mathcal{H},\sigma,\rho,1}$ is a positive constant independent of m or δ and will be given explicitly in the proof.

Proof For any $g \in \mathcal{F}_\mathcal{H}$, we know from the definition of $\mathcal{F}_\mathcal{H}$ that g can be expressed as

$$g(z) = -\sigma^2 \exp\left\{-(y - f(x))^2/\sigma^2\right\} + \sigma^2 \exp\left\{-(y - f_\rho(x))^2/\sigma^2\right\}, z \in \mathcal{Z},$$

for some $f \in \mathcal{H}$. Following from the proof of Proposition 10, we know that

$$\|g\|_\infty \leq \sqrt{2/e}\sigma \sup_{f\in\mathcal{H}} \|f - f_\rho\|_\infty := B_{\mathcal{H},\sigma,\rho}.$$

When the Noise Assumption holds, $f_\rho \in \mathcal{H}$, and $\sigma > \sigma_{\mathcal{H},\rho}$, we have

$$\begin{aligned}
\mathbb{E}g^2 &\leq \mathbb{E}\left((f(x) - f_\rho(x))^2(2y - f(x) - f_\rho(x))^2\right) \\
&\leq \int_\mathcal{Y} \Big(12y^2 + 3\sup_{f\in\mathcal{H}} \|f\|^2_\infty + 3\|f_\rho\|^2_\infty\Big) d\rho(y|x) \int_\mathcal{X} (f(x) - f_\rho(x))^2 d\rho_\mathcal{X}(x) \\
&\leq c^{-1}_{\mathcal{H},\sigma,\rho}\Big(12\int_\mathcal{Z} y^2 d\rho + 3\sup_{f\in\mathcal{H}} \|f\|^2_\infty + 3\|f_\rho\|^2_\infty\Big)\mathbb{E}g := c_{\mathcal{H},\sigma,\rho,1}\mathbb{E}g,
\end{aligned}$$

where the last inequality follows from Lemma 18. For any $g_1,\, g_2 \in \mathcal{F}_{\mathcal{H}}$, there exist $f_1,\, f_2 \in \mathcal{H}$ such that

$$g_1(z) = -\sigma^2 \exp\left\{-(y - f_1(x))^2/\sigma^2\right\} + \sigma^2 \exp\left\{-(y - f_\rho(x))^2/\sigma^2\right\}$$

and

$$g_2(z) = -\sigma^2 \exp\left\{-(y - f_2(x))^2/\sigma^2\right\} + \sigma^2 \exp\left\{-(y - f_\rho(x))^2/\sigma^2\right\}.$$

From the proof of Proposition 10, we know that $|g_1 - g_2| \leq \sqrt{2/e}\sigma\|f_1 - f_2\|_\infty$. This in connection with the Complexity Assumption II yields

$$\log \mathcal{N}_2(\mathcal{F}_{\mathcal{H}}, \eta) \leq \log \mathcal{N}_2\left(\mathcal{H}, \eta/(\sqrt{2/e}\sigma)\right) \leq c_{II,s}\left(\sqrt{2/e}\sigma/\eta\right)^s := a_{\sigma,s}\eta^{-s}.$$

Applying Lemma 14 to the function set $\mathcal{F}_{\mathcal{H}}$, with simple computations, we see that for any $0 < \delta < 1$ with confidence $1 - \delta$, there holds

$$\{\mathcal{E}^\sigma(f) - \mathcal{E}^\sigma(f_\rho)\} - \{\mathcal{E}^\sigma_{\mathbf{z}}(f) - \mathcal{E}^\sigma_{\mathbf{z}}(f_\rho)\} \leq \frac{1}{2}\{\mathcal{E}^\sigma(f) - \mathcal{E}^\sigma(f_\rho)\} + C_{\mathcal{H},\sigma,\rho,1}\log(1/\delta)m^{-2/(2+s)},$$

where $C_{\mathcal{H},\rho,\sigma,1}$ is a positive constant independent of m or δ and given by

$$C_{\mathcal{H},\rho,\sigma,1} = 18B_{\mathcal{H},\sigma,\rho} + 2c_{\mathcal{H},\sigma,\rho,1} + 2a'_s a^{2/(2+s)}_{\sigma,s}(c_{\mathcal{H},\sigma,\rho,1} + B_{\mathcal{H},\sigma,\rho})^{(2-s)/(2+s)},$$

and a'_s is a positive constant depending only on s. This completes the proof of Proposition 19. ■

Proof [Proof of Theorem 6] As a consequence of Proposition 19, we see that for any $0 < \delta < 1$, with confidence $1 - \delta$, there holds

$$\mathcal{S}_2(\mathbf{z}) \leq \frac{1}{2}\{\mathcal{E}^\sigma(f_{\mathbf{z}}) - \mathcal{E}^\sigma(f_\rho)\} + C_{\mathcal{H},\sigma,\rho,1}\log(1/\delta)m^{-2/(2+s)}.$$

Following from Lemma 18 and recalling that $\mathcal{S}_1(\mathbf{z}) = 0$, we come to the conclusion that for any $0 < \delta < 1$, with confidence $1 - \delta$, there holds

$$\|f_{\mathbf{z}} - f_\rho\|^2_{\mathcal{L}^2_{\rho_{\mathcal{X}}}} \leq c^{-1}_{\mathcal{H},\sigma,\rho}\{\mathcal{E}^\sigma(f_{\mathbf{z}}) - \mathcal{E}^\sigma(f_\rho)\} = c^{-1}_{\mathcal{H},\sigma,\rho}\mathcal{S}_2(\mathbf{z}) \leq 2c^{-1}_{\mathcal{H},\sigma,\rho}C_{\mathcal{H},\sigma,\rho,1}\log(1/\delta)m^{-2/(2+s)}.$$

By denoting $C_{\mathcal{H},\sigma,\rho} = 2c^{-1}_{\mathcal{H},\sigma,\rho}C_{\mathcal{H},\sigma,\rho,1}$, we complete the proof of Theorem 6. ■

5. Towards the Role that σ Plays

We now move our attention to discuss the scale parameter σ in the ℓ_σ loss by making some attempts to interpret the role that σ plays from a learning theory viewpoint.

The first observation on the parameter σ in the ℓ_σ loss is that it determines the robustness of the regression models. For linear regression models, this observation has been quantitatively described in terms of the influence function and finite-sample breakdown

point in Wang et al. (2013). For nonlinear regression models, similar observations on the robustness have been also empirically reported. For instance, the robustness of the regression models induced by the ℓ_σ losses can be enhanced with a decreasing value of σ. In fact, this is reasonable if we look at the ℓ_σ loss in which a smaller σ would limit the influence of the outliers in the response variable. In addition, in the learning theory literature, the robustness property of kernel-based regression models has been studied by considering the growth type of the loss function and investigating the existence and boundedness of the corresponding influence function (see Christmann and Steinwart, 2007; Steinwart and Christmann, 2008). From Chapter 2 in Steinwart and Christmann (2008), it is easy to check that the ℓ_σ loss is of upper growth type 1 due to its Lipschitz continuity property and consequently can be used to deal with unbounded Y. It would be also worthwhile to derive a quantitative description on the robustness of the MCCR model (2) in terms of the influence function as done in Christmann and Steinwart (2007) and Christmann and Messem (2008) for convex regression models. However, we remark that due to the non-convexity of the ℓ_σ loss, the deduction of the influence function of the MCCR model in $\mathcal{H}$ (which is possibly infinite dimensional) can be much involved and is worthy for further study.

On the other hand, we realize that in the robustness literature, the scale parameter not only controls the robustness property of the regression model associated with the ℓ_σ loss but also specifies its efficiency and plays a trade-off role. Considering the nonparametric setting in our study and given that our primary concern is the convergence rates of the MCCR model (2), we restrict ourselves to discussions of the influence of the scale parameter σ on the convergence rates. To this end, we recall the following relation from the error decomposition in Proposition 9:

$$\|f_{\mathbf{z}} - f_\rho\|^2_{\mathcal{L}^2_{\rho_\mathcal{X}}} \leq \left\{\mathcal{E}^\sigma(f_{\mathbf{z}}) - \mathcal{E}^\sigma(f^\sigma_\mathcal{H})\right\} + \|f_\mathcal{H} - f_\rho\|^2_{\mathcal{L}^2_{\rho_\mathcal{X}}} + \mathcal{A}_{\mathcal{H},\sigma,\rho}.$$

On the right-hand side of the above inequality, the first term is the excess risk of the empirical estimator modeled by the MCCR model, the convergence of which can be ensured by controlling the complexity of the hypothesis space $\mathcal{H}$ and confining the tail behavior of the response variable. The second term $\|f_\mathcal{H} - f_\rho\|^2_{\mathcal{L}^2_{\rho_\mathcal{X}}}$ represents the approximation error and is independent of the scale parameter σ. The influence of the scale parameter σ on the convergence rates can be revealed from the bias term $\mathcal{A}_{\mathcal{H},\sigma,\rho}$. According to Proposition 9, we know that $\mathcal{A}_{\mathcal{H},\sigma,\rho} = 2c_{\mathcal{H},\rho}/\sigma^2$. Therefore, a decreasing value of σ will lead to increasing bias and consequently yields slower convergence rates.

From the above discussions, we can see that the parameter σ in the ℓ_σ loss balances the robustness of the MCCR model (2) and its convergence rates. We will continue our discussion on the role that σ plays by trying to extend our preceding analysis for the ℓ_σ loss to other robust regression loss functions in the next section.

6. Generalization to Other Robust Loss Functions

In the preceding sections, motivated by the information-theoretic interpretation of the maximum correntropy criterion and its empirical successes in real-world applications, we generalize the idea of the maximum correntropy criterion in regression with the ℓ_σ loss. We then present a theoretical understanding towards the maximum correntropy criterion in regression by conducting a learning theory analysis for $\|f_{\mathbf{z}} - f_\rho\|^2_{\mathcal{L}^2_{\rho_\mathcal{X}}}$. We conclude that one can

rely on the ℓ_σ loss to solve regression problems with non-Gaussian as well as Gaussian noise. However, one may argue that from a regression viewpoint, the ℓ_σ loss is merely a special case of robust loss functions arise in robust statistics. In view of this, in this section we try to generalize our previous analysis to other robust loss functions and see what happens when a robust loss function is applied into the learning for regression scenarios.

The robust loss functions refer to those used to obtain robust M-estimators in linear regression models. As mentioned earlier, the MCCR model can be viewed as a nonparametric M-estimator. Therefore, we first give a glimpse of the robust M-estimation methods in linear regression models to distinguish them from the robust nonparametric M-estimator we investigate in this paper. In linear regression models, it is assumed that the observations $\mathbf{z}$ are drawn i.i.d from $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ with $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{Y} = \mathbb{R}$. In this setting, the regression function $f^\star(x) := x^T\theta^\star$, where $\theta^\star \in \Theta := \mathbb{R}^d$ is unknown and one of the main tasks in linear regression problem is to estimate the regression parameter $\theta^\star$. A common approach to obtaining a robust estimator $\hat{\theta}$ for $\theta^\star$ is to solve the following optimization problem

$$\hat{\theta} = \arg\min_{\theta\in\mathbb{R}^d} \sum_{i=1}^m \phi\left(\frac{y_i - x_i^T\theta}{\sigma}\right), \tag{19}$$

where $\sigma > 0$ is the scale parameter and ϕ is a robust loss function that downweights large residual errors. In fact, by using the above robust loss function ϕ, concerning the nonlinear regression model (1), one can also propose the following robust nonparametric ERM-based regression scheme

$$\hat{f}_{\mathbf{z}} = \arg\min_{f\in\mathcal{H}} \sum_{i=1}^m \phi\left(\frac{y_i - f(x_i)}{\sigma}\right). \tag{20}$$

Notice that (19) aims at estimating a vector in $\mathbb{R}^d$ while (20) is proposed to estimate a function in a function space $\mathcal{H}$ that can have an infinite dimension. This gives the main difference between the two models. Denoting $\phi_\sigma(t) := \phi(t/\sigma)$, besides the ℓ_σ loss investigated in this paper, several frequently employed robust loss functions include:

- Huber's loss: $\phi_\sigma(t) = t^2 I_{\{|t|\le\sigma\}} + (2\sigma|t| - \sigma^2) I_{\{|t|>\sigma\}}$;
- Cauchy loss: $\phi_\sigma(t) = \sigma^2 \log\left(1 + t^2/\sigma^2\right)$;
- Tukey's biweight loss: $\phi_\sigma(t) = (\sigma^2/6)(1 - (1 - (t/\sigma)^2)^3) I_{\{|t|\le\sigma\}} + (\sigma^2/6) I_{\{|t|>\sigma\}}$.

In the above loss functions, I_S is an indicator function which takes the value 1 if S is true and gets the value 0 otherwise.

Recall that our previous analysis on the ℓ_σ loss and the MCCR model (2) relies heavily on Lemma 7. From the proof of Lemma 7, we know that similar analysis can be also applied to other robust loss functions that are sufficiently smooth and satisfy certain conditions, e.g., the Cauchy loss given above. On the other hand, although our analysis cannot cover all the robust loss functions, following from our previous analyzing process, we can still get a general view on the robust loss functions and see what happens when a robust loss function is employed from a learning theory viewpoint.

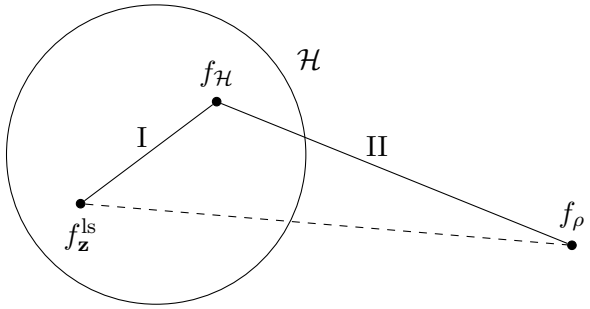


Figure 2: The statistical learning approach to bounding the $\mathcal{L}^2_{\rho_{\mathcal{X}}}$-distance between $f_{\mathbf{z}}$ and f_ρ for the ERM scheme (6), which is induced by the least squares loss.

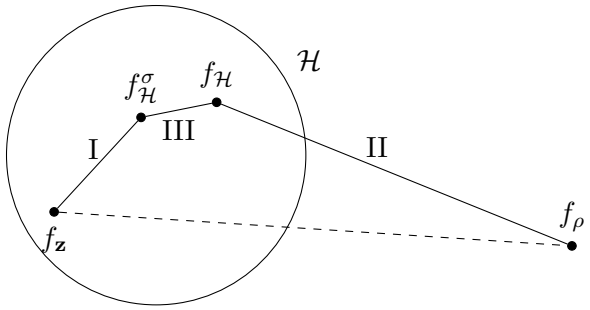


Figure 3: The statistical learning approach to bounding the $\mathcal{L}^2_{\rho_{\mathcal{X}}}$-distance between $f_{\mathbf{z}}$ and f_ρ for the ERM scheme induced by a robust loss function ϕ_σ.

To illustrate this, we first recall that to analyze the convergence of an ERM scheme associated with the least squares loss (e.g., the unconstrained regression model (6)), a typical statistical learning approach is proceeded as follows: instead of directly measuring the $\mathcal{L}^2_{\rho_\mathcal{X}}$-distance between $f^{\rm ls}_{\mathbf{z}}$ and f_ρ, one first introduces the projection of f_ρ in $\mathcal{H}$, i.e., $f_\mathcal{H}$. With the help of $f_\mathcal{H}$, one can decompose the distance into sample error and approximation error as follows:

$$\|f^{\rm ls}_{\mathbf{z}} - f_\rho\|^2_{\mathcal{L}^2_{\rho_\mathcal{X}}} \leq \|f^{\rm ls}_{\mathbf{z}} - f_\mathcal{H}\|^2_{\mathcal{L}^2_{\rho_\mathcal{X}}} + \|f_\mathcal{H} - f_\rho\|^2_{\mathcal{L}^2_{\rho_\mathcal{X}}}.$$

The idea of the above decomposition is depicted in Figure 2, where I represents the sample error $\|f^{\rm ls}_{\mathbf{z}} - f_\mathcal{H}\|^2_{\mathcal{L}^2_{\rho_\mathcal{X}}}$ while II gives the approximation error $\|f_\mathcal{H} - f_\rho\|^2_{\mathcal{L}^2_{\rho_\mathcal{X}}}$.

However, situations will be quite different if a robust regression loss ϕ_σ is employed. To explain this, we redefine $f^\sigma_\mathcal{H}$ as the target function of the regression model induced by a general robust loss ϕ_σ and $f_{\mathbf{z}}$ as the corresponding empirical target function, definitions of which are given as follows

$$f^\sigma_\mathcal{H} = \arg\min_{f\in\mathcal{H}} \int_\mathcal{Z} \phi_\sigma(y - f(x))d\rho \text{ and } f_{\mathbf{z}} = \arg\min_{f\in\mathcal{H}} \frac{1}{m}\sum_{i=1}^m \phi_\sigma(y_i - f(x_i)).$$

The analysis in our study indicates that to analyze the convergence of a regression model induced by a robust loss function ϕ_σ, one may proceed via the following decomposition

$$\|f_{\mathbf{z}} - f_\rho\|^2_{\mathcal{L}^2_{\rho_\mathcal{X}}} \leq \|f_{\mathbf{z}} - f^\sigma_\mathcal{H}\|^2_{\mathcal{L}^2_{\rho_\mathcal{X}}} + \|f^\sigma_\mathcal{H} - f_\mathcal{H}\|^2_{\mathcal{L}^2_{\rho_\mathcal{X}}} + \|f_\mathcal{H} - f_\rho\|^2_{\mathcal{L}^2_{\rho_\mathcal{X}}}.$$

Figure 3 gives an intuitive description on the above decomposition. Similarly, in Figure 3, I represents the sample error term $\|f_{\mathbf{z}} - f^\sigma_\mathcal{H}\|^2_{\mathcal{L}^2_{\rho_\mathcal{X}}}$, II stands for the approximation error term $\|f_\mathcal{H} - f_\rho\|^2_{\mathcal{L}^2_{\rho_\mathcal{X}}}$ while III measures the $\mathcal{L}^2_{\rho_\mathcal{X}}$-distance between $f^\sigma_\mathcal{H}$ and $f_\mathcal{H}$. Notice that the bias term III is caused by the introduction of the scale parameter σ that delivers the robustness to the model. Due to the non-robustness of LSR and the fact that $f_\mathcal{H}$ is the target function of LSR, again we conclude that the smaller of the $\mathcal{L}^2_{\rho_\mathcal{X}}$-distance between $f^\sigma_\mathcal{H}$ and $f_\mathcal{H}$ is, the less robustness the regression model associated with the ϕ_σ loss possesses.

Taking the ℓ_σ loss for example, we know from our previous analysis that under very specific conditions the two points, $f^\sigma_\mathcal{H}$ and $f_\mathcal{H}$, meet and consequently the bias term III disappears. Technically speaking, a nice point of the ℓ_σ loss lies in that it is sufficiently smooth which makes it possible to bound the $\mathcal{L}^2_{\rho_\mathcal{X}}$-distance between $f^\sigma_\mathcal{H}$ and $f_\mathcal{H}$ explicitly. For instance, when the Moment Assumption holds and $f_\rho \in \mathcal{H}$, as a consequence of Lemma 7, we see that

$$\|f^\sigma_\mathcal{H} - f_\mathcal{H}\|^2_{\mathcal{L}^2_{\rho_\mathcal{X}}} \leq c_{\mathcal{H},\rho}/\sigma^2.$$

As mentioned in the previous section, the above estimate reveals that when the value of σ decreases, the upper bound of the bias term III increases.

Based on the above discussions, we conclude that when a robust loss function is employed in nonparametric regression problems, the enhancement of robustness is at the sacrifice of the convergence rate of the model and what one needs to do is to find a good compromise.

7. Numerical Experiments

Studies in this paper are motivated by empirical success of the MCCR model. However, for the sake of completeness, in this section, we carry out numerical experiments on synthetic and real data sets to show the effectiveness of the MCCR model (2).

7.1 Experimental Setup

Notice that the MCCR model (2) is a constrained optimization model since $\mathcal{H}$ is assumed to be a compact subset of $C(\mathcal{X})$. As mentioned previously, a typical choice of $\mathcal{H}$ is a bounded subset of a certain reproducing kernel Hilbert space $\mathcal{H}_K$ induced by some Mercer kernel K. However, to determine the diameter of this bounded subset in applications, prior information is usually required. In our experiments, instead of evaluating the optimization model (2), we focus on its unconstrained version

$$f_{\mathbf{z}} = \arg\min_{f\in\mathcal{H}_K} \frac{1}{m}\sum_{i=1}^{m} \ell_\sigma(y_i, f(x_i)) + \lambda\|f\|_K^2, \tag{21}$$

where λ is a positive regularization parameter.

The representer theorem ensures that we can search within the function set $\mathcal{H}_{K,\mathbf{z}}$ for the minimizer of the optimization model (21), where

$$\mathcal{H}_{K,\mathbf{z}} = \left\{ \sum_{i=1}^{m} \alpha_i K(x, x_i) + b,\, b\in\mathbb{R}, \alpha_i\in\mathbb{R},\, i = 1, \cdots, m \right\},$$

with b being an offset. In our experiments, we use the Gaussian kernel

$$K_h(x_i, x_j) = \exp\left(-\|x_i - x_j\|^2/h^2\right),$$

with the parameter h to be determined. To show the effectiveness of the MCCR model, we compare the empirical performance of (21) with other robust regression schemes, including robust regression models based on the Huber's loss and the least absolute deviation loss. These robust regression schemes are obtained by replacing the ℓ_σ loss in (21) with the Huber's loss and the least absolute deviation loss, respectively. Explicit definitions of the two loss functions are given as follows:

$$\phi_a^{\text{Huber}}(u,v) = \begin{cases} (u-v)^2, & \text{if } |u-v|\le a \\ 2a|u-v|, & \text{if } |u-v| > a \end{cases} \quad \text{and} \quad \phi^{\text{LAD}}(u,v) = |u-v|,\ u,v\in\mathbb{R}.$$

For notation simplifications, we denote the two robust regression models as Huber and LAD, respectively.

To solve (21), we apply the iteratively reweighted least squares method (IRLS). The basic procedure is to iteratively solve the weighted least squares problem and give weights according to the current solution. Due to the non-convexity of the MCCR model, solving (21) by using IRLS only guarantees a stationary point. In our experiment, we use the result of the least squares method as the starting point.

In our experiment, noise added to the toy examples is given as follows

$$\text{noise} := \tau_1\varepsilon_1 + \tau_2\varepsilon_2^p, \tag{22}$$

where ε_1 follows the standard Gaussian distribution and ε_2^p is an impulse noise (outliers) defined as

$$\text{Prob}(\varepsilon_2^p = t) = \begin{cases} 1-p, & t=0, \\ p/2, & t=1, \\ p/2, & t=-1. \end{cases}$$

τ_1 and τ_2 are introduced to set the variance of the Gaussian noise and the magnitude of the impulsive noise. In our experiment, we always set $p = 0.1$, i.e., 10% samples are contaminated by impulsive noise. In addition, in some of our experiments on synthetic data sets, we will also consider the noise ε_1 that is drawn from the Student's t-distribution with 3 degrees of freedom, and Cauchy distribution.

7.2 Example of the Noisy Sinc Function

We first choose the sinc function as the regression function. The one-dimension sinc function is given as

$$f(x) = \sin(\pi x)/(\pi x), \quad x \in [-4, 4], \tag{23}$$

which is frequently adopted to illustrate the regression models (see Vapnik, 1998; Suykens et al., 2002a,b; Schölkopf et al., 2000; Smola and Schölkopf, 2004).

In our experiment, we first draw a training set of size 100 from the sinc function (23) that are corrupted by the Gaussian noise. We then draw another training set with the same size corrupted by the Gaussian noise and the outliers. With each training set, the fitting results of the sinc function are plotted in Figure 4, in which the red dot-dashed curve is the one reconstructed by MCCR, the blue dashed curve represents the one from Huber while LAD gives the green dotted curve.

From Figure 4, one can see that all of the three models can fit the curve of the sinc function well when the data is only contaminated by the Gaussian noise. When the training data are also corrupted by outliers, all of the three robust regression models can still successfully reconstruct the curve. However, we can see that MCCR gives the best fitting results, especially at positions where data are corrupted by outliers.

7.3 Example of the Noisy Friedman's Benchmark Functions

Our second numerical experiment on toy examples considers multiple dimensional regression problems. We now use the Friedman's benchmark functions as our test functions, which were introduced in Friedman (1991) and have become widely employed models when studying regression problems (see Tipping, 2001; Brown et al., 2005; Debruyne et al., 2010).

The Friedman's benchmark functions are listed as follows:

- $f_1(x) = 10\sin(\pi x^1 x^2) + 20(x^3 - 0.5)^2 + 10x^4 + 5x^5$;
- $f_2(x) = \sqrt{(x^1)^2 + \left(x^2 x^3 - 1/(x^2 x^4)\right)^2}$;
- $f_3(x) = \arctan\left(1/x^1 \left(x^2 x^3 - 1/(x^2 x^4)\right)\right)$.

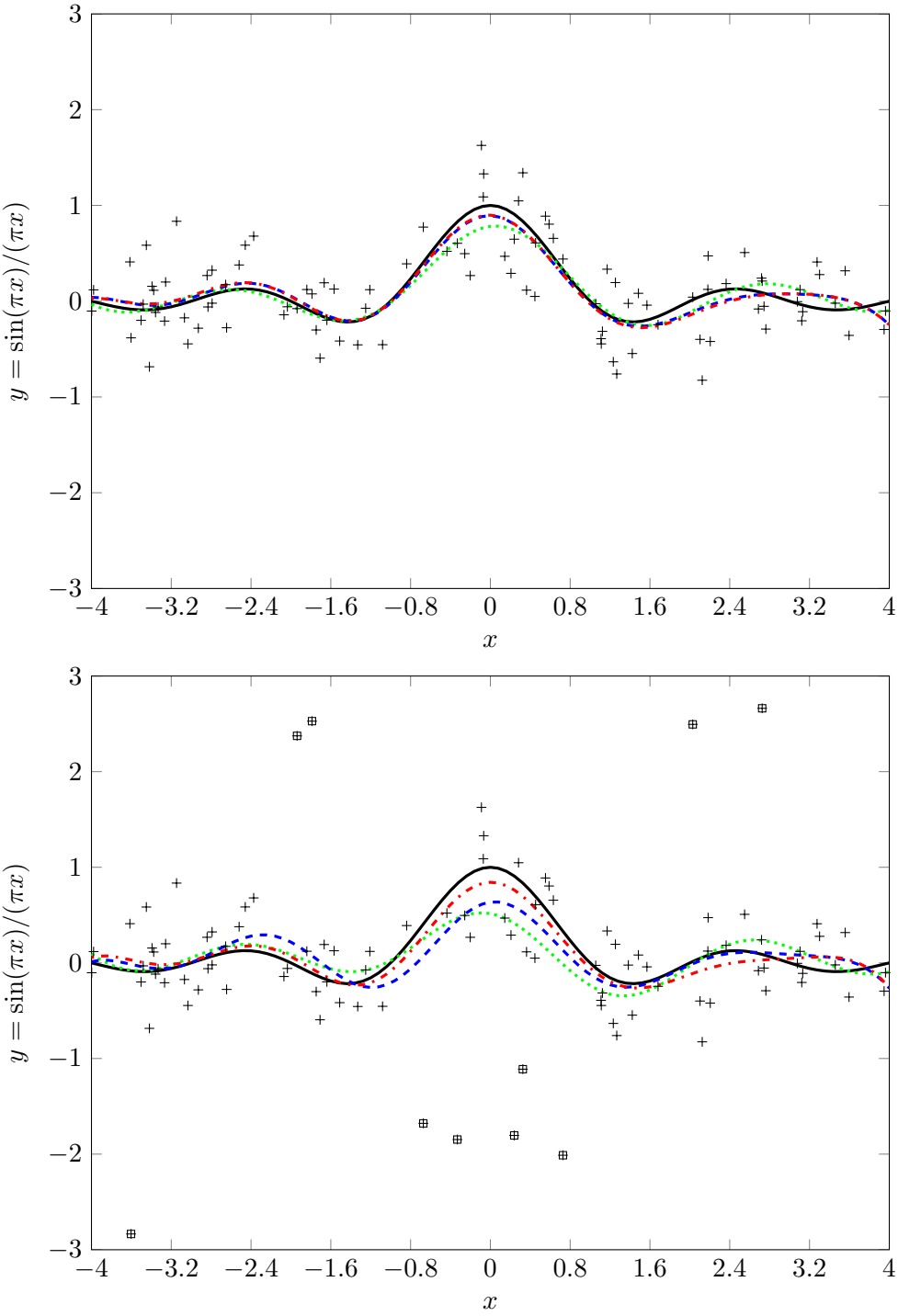


Figure 4: Sinc function (black solid curves) and the regression results (MCCR: red dot-dashed curve; Huber: blue dashed curve; LAD: green dotted curve). (top) The training data (crosses) are corrupted by Gaussian noise; (bottom) Some observed data are outliers (marked by squares).

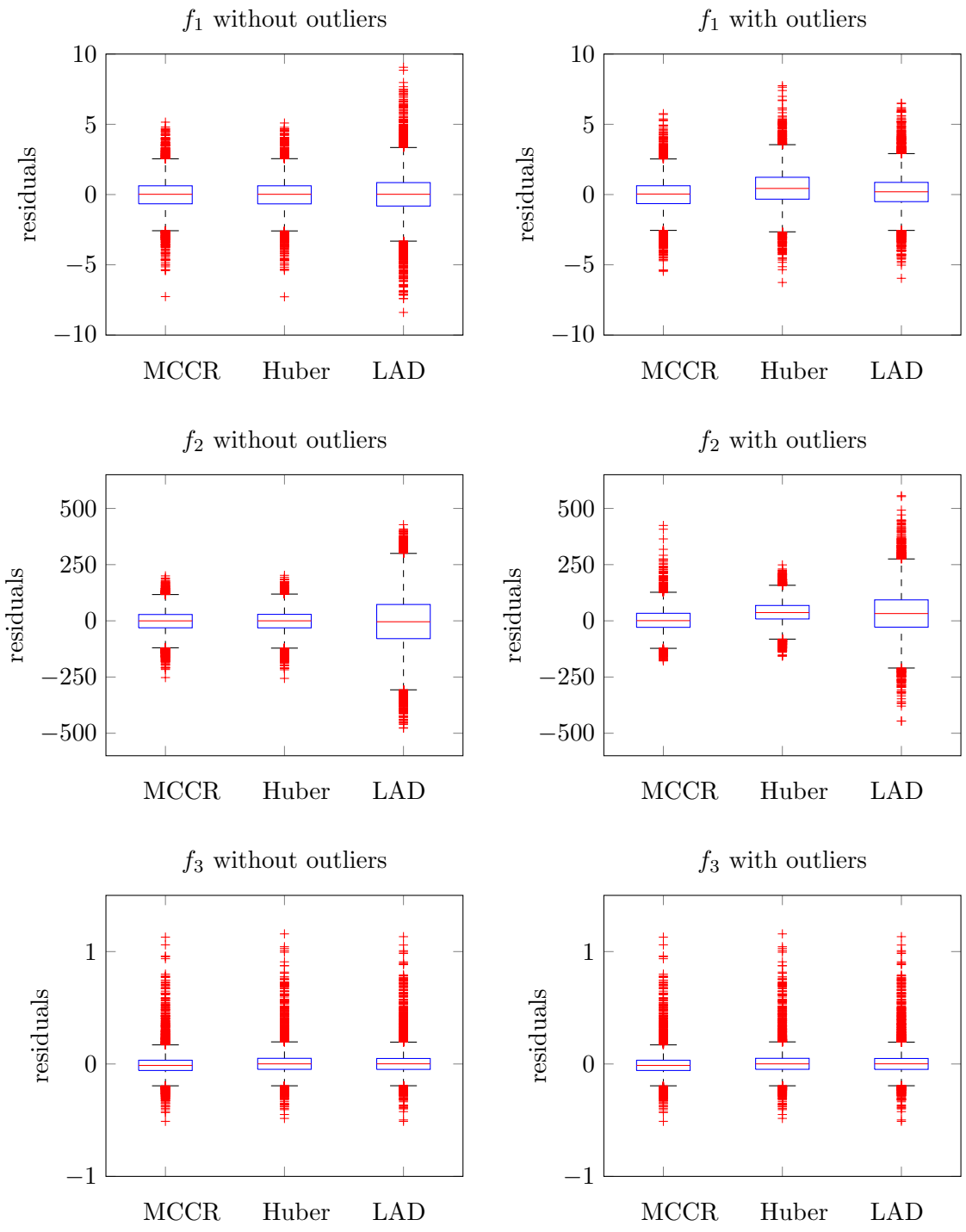


Figure 5: Box-plots of the residuals of Friedman's benchmark functions for the case of Gaussian noise. Each box-plot features a lower quartile (25 percentile) line, a median (50 percentile) line and an upper quartile (75 percentile) line for the residuals on test data.

For f_1, $x = (x^1, \ldots, x^{10})$ where each x^j, $j = 1, \ldots, 10$, is uniformly distributed in $[0, 1]$ and x^6, $\ldots$, x^{10} are noisy variables. For f_2 and f_3, $x = (x^1, x^2, x^3, x^4)$ and each is uniformly distributed in the following intervals: $x^1 \in [0, 100]$, $x^2 \in [40\pi, 560\pi]$, $x^3 \in [0, 1]$ and $x^4 \in [1, 11]$.

For each function, 1000 observations are randomly taken from corresponding domain for training and cross-validating. Another independent 1000 observations are also randomly drawn as the test set. Noise and outliers are then added according to (22). For f_1, we set $\tau_1 = 1$. For f_2 and f_3, τ_1 is set such that the ratio of the signal power to the power of ε_1 is 3. In the outlier-free cases, we set $\tau_2 = 0$. To observe the performance for the three models in the presence of outliers in the training data sets, we set $\tau_2 = \max_{x \in D} f(x) - \min_{x \in D} f(x)$, where D is the domain of each benchmark function. For each regression model, the width of the Gaussian kernel h, the regularization parameter λ and the scale parameter in the loss function (no scale parameter for the LAD loss) are all tuned via a 10-fold cross-validation under the mean squared error criterion. The residuals $\{y_i - f(x_i)\}_{i=1}^{1000}$ are recorded. For the case of Gaussian noise, we boxplot all the residuals in Figure 5. Each box-plot features a lower quartile (25 percentile) line, a median (50 percentile) line and an upper quartile (75 percentile) line.

In Table 1, we also report the relative sum of squared error (RSSE) on the test data set T, i.e.,

$$\mathrm{RSSE}(\hat{f}) = \sum_{x \in T} \Big(f(x) - \hat{f}(x) \Big)^2 \Big/ \sum_{x \in T} \big(f(x) - \bar{f}_T \big)^2 ,$$

where $\bar{f}_T$ is the mean value of $f(x)$ on T.

test function	noise	MCCR	Huber	LAD
f_1	Gaussian noise, no outliers	**0.048**	0.049	0.103
	Gaussian noise, outliers	**0.062**	0.073	0.157
f_2	Gaussian noise, no outliers	**0.020**	0.021	0.136
	Gaussian noise, outliers	**0.023**	0.032	0.156
f_3	Gaussian noise, no outliers	**0.091**	0.117	0.136
	Gaussian noise, outliers	**0.062**	0.073	0.157
f_1	Cauchy noise, no outliers	**0.042**	0.042	0.116
	Cauchy noise, outliers	**0.045**	0.049	0.089
f_2	Cauchy noise, no outliers	**0.005**	0.005	0.025
	Cauchy noise, outliers	**0.006**	0.006	0.021
f_3	Cauchy noise, no outliers	0.180	0.195	**0.177**
	Cauchy noise, outliers	0.219	**0.143**	0.154
f_1	Student noise, no outliers	**0.040**	0.040	0.101
	Student noise, outliers	**0.046**	0.075	0.092
f_2	Student noise, no outliers	**0.017**	0.017	0.129
	Student noise, outliers	**0.023**	0.024	0.123
f_3	Student noise, no outliers	**0.423**	0.429	0.430
	Student noise, outliers	0.471	0.544	**0.434**

Table 1: The relative sum of squared error on the test data

7.4 Evaluation on Real Data Sets

We also evaluate the three robust regression models on four real data sets downloaded from UCI repository of machine learning databases (see Bache and Lichman, 2013): Concrete Compressive Strength Data Set, Housing Data Set, Yacht Hydrodynamics Data Set and Airfoil Self-Noise Data Set.

For each data set, two third of the instances are used for training and the remaining are used for test. We repeat our experiment as done for the Friedman's benchmark functions for ten times. The residuals for the three robust regression models are displayed by box-plots in Figure 6, the accuracy of which are measured by RSSE. Experimental results on the RSSEs and the details of training data, including the size of features n and the size of instances m, are reported in Table 2.

data sets	n	m	MCCR	Huber	LAD
concrete	9	686	**0.061**	0.061	0.062
house	14	338	0.128	**0.126**	0.175
yacht-hydrodynamics	7	205	**0.022**	0.024	0.159
airfoil	6	1000	**0.184**	0.195	0.238

Table 2: The relative sum of squared error on real data

In the above numerical evaluations on toy examples and real data sets, our experiments show that when the data is only contaminated by Gaussian noise, a large sigma value in the MCCR model and a large a value in the regression model based on the Huber's criterion will be selected via cross-validation. However, for other noise and in the presence and absence of outliers, smaller values of the scale parameters in the two regression models will be selected. These coincide with our understandings on the robust regression models.

From the above experimental results, we can see the effectiveness of MCCR especially for the cases in the presence of impulsive noise.

8. Concluding Remarks

In this paper, we presented a statistical learning interpretation of the regression model associated with the correntropy induced regression loss. We investigated its connections with the least squares regression. We found that the correntropy induced loss could help for regression with non-Gaussian noise. Meanwhile, comparable performance could be obtained by applying this regression model when the noise is Gaussian. Convergence rates of the proposed model under various circumstances were derived explicitly. We showed that the scale parameter in the loss function balanced the convergence rates and the robustness of the model. We also made some efforts to extend our analysis to other robust loss functions and gave a general view on analyzing regression models induced by general robust loss functions. It is expected that our observations can shed some light towards future real-life applications.

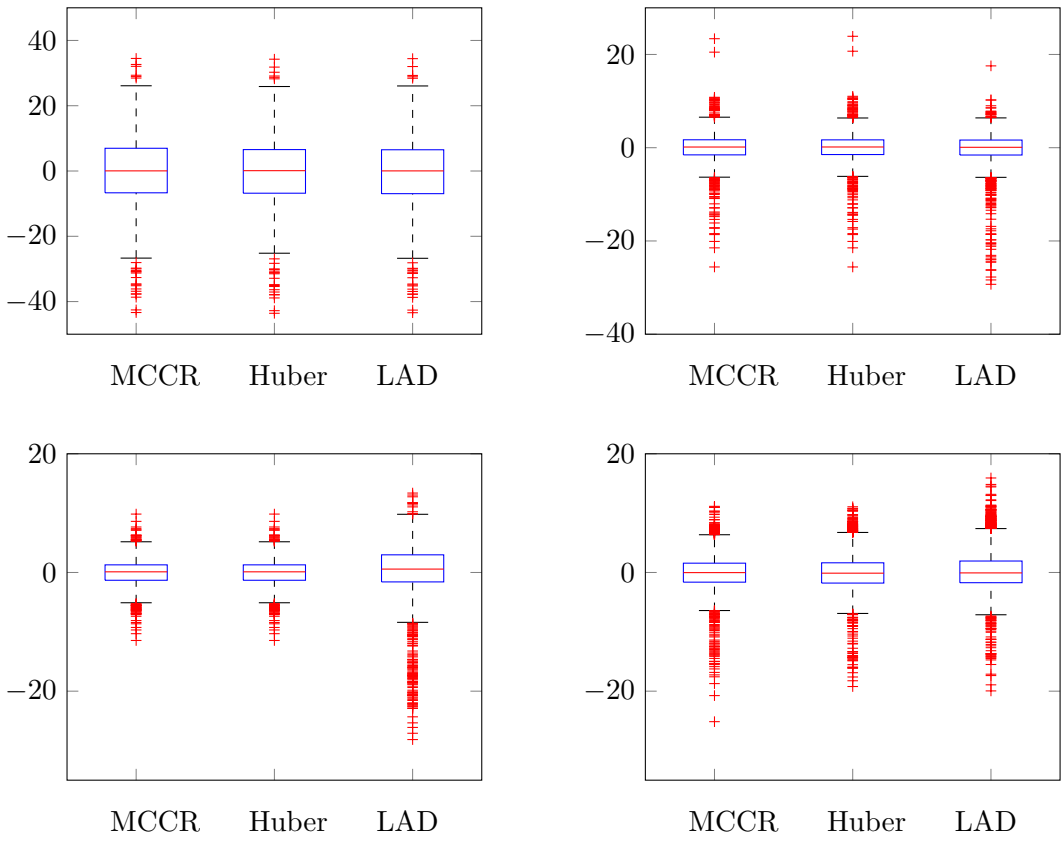


Figure 6: Box-plots of the residuals on four real data sets. Each box-plot features a lower quartile (25 percentile) line, a median (50 percentile) line and an upper quartile (75 percentile) line for the residuals on test data. (top left) concrete; (top right) Boston house; (bottom left) yacht hydrodynamics; (bottom right) airfoil.

Acknowledgments

The authors would like to thank the action editor and the reviewers for their insightful comments and constructive suggestions, which improved the quality of this paper. The authors would also like to thank Dr. Jun Fan from Department of Statistics, University of Wisconsin-Madison for helpful discussions.

EU: The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC AdG A-DATADRIVE-B (290923). This paper reflects only the authors' views, the Union is not liable for any use that may be made of the contained information. Research Council KUL: GOA/10/09 MaNet, CoE PFV/10/002 (OPTEC), BIL12/11T; PhD/Postdoc grants. Flemish Government: FWO: projects: G.0377.12 (Structured systems), G.088114N (Tensor based data similarity); PhD/Postdoc grants. IWT: projects: SBO POM (100031); PhD/Postdoc grants. iMinds Medical Information Technologies SBO 2014. Belgian Federal Science Policy Office: IUAP P7/19 (DYSCO, Dynamical systems, control and optimization, 2012-2017). L. Shi is supported by the National Natural Science Foundation of China Project No. 11201079), the Joint Research Fund by National Natural Science Foundation of China and Research Grants Council of Hong Kong (Project No. 11461161006 and Project No. CityU 104012) and the Fundamental Research Funds for the Central Universities of China (Project No. 20520133238, Project No. 20520131169). Johan Suykens is a professor at KU Leuven, Belgium.

References

Martin Anthony and Peter L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, Cambridge, 1999.

Jean-Yves Audibert and Olivier Catoni. Robust linear least squares regression. *Annals of Statistics*, 39(5):2766–2794, 2011.

Kevin Bache and Moshe Lichman. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml`.

Sergei N. Bernstein. *The Theory of Probabilities*. Gastehizdat Publishing House, Moscow, 1946.

Ricardo J. Bessa, Vladimiro Miranda, and João Gama. Entropy and correntropy against minimum square error in offline and online three-day ahead wind power forecasting. *Power Systems, IEEE Transactions on*, 24(4):1657–1666, 2009.

Gavin Brown, Jeremy L. Wyatt, and Peter Tiňo. Managing diversity in regression ensembles. *Journal of Machine Learning Research*, 6:1621–1650, 2005.

Andrea Caponnetto and Ernesto De Vito. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7(3):331–368, 2007.

Andreas Christmann and Arnout Van Messem. Bouligand derivatives and robustness of support vector machines for regression. *Journal of Machine Learning Research*, 9:915–936, 2008.

Andreas Christmann and Ingo Steinwart. Consistency and robustness of kernel-based regression in convex risk minimization. *Bernoulli*, 13(3):799–819, 2007.

Felipe Cucker and Ding-Xuan Zhou. *Learning Theory: An Approximation Theory Viewpoint*. Cambridge University Press, Cambridge, 2007.

Patrick L. Davies. Aspects of robust linear regression. *Annals of Statistics*, 21(4):1843–1899, 1993.

Kris De Brabanter, Kristiaan Pelckmans, Jos De Brabanter, Michiel Debruyne, Johan A. K. Suykens, Mia Hubert, and Bart De Moor. Robustness of kernel based regression: a comparison of iterative weighting schemes. In *International Conference on Artificial Neural Networks–ICANN 2009*, pages 100–110. Springer, 2009.

Michiel Debruyne, Mia Hubert, and Johan A. K. Suykens. Model selection in kernel based regression using the influence function. *Journal of Machine Learning Research*, 9:2377–2400, 2008.

Michiel Debruyne, Andreas Christmann, Mia Hubert, and Johan A. K. Suykens. Robustness of reweighted least squares kernel based regression. *Journal of Multivariate Analysis*, 101 (2):447–463, 2010.

John E. Dennis and Roy E. Welsch. Techniques for nonlinear least squares and robust regression. *Communications in Statistics-Simulation and Computation*, 7(4):345–359, 1978.

Jun Fan, Ting Hu, Qiang Wu, and Ding-Xuan Zhou. Consistency analysis of an empirical minimum error entropy algorithm. *Applied and Computational Harmonic Analysis, in press*. doi: 10.1016/j.acha.2014.12.005.

Jerome H. Friedman. Multivariate adaptive regression splines. *Annals of Statistics*, 19(1): 1–141, 1991.

Aysegul Gunduz and José C. Príncipe. Correntropy as a novel measure for nonlinearity tests. *Signal Processing*, 89(1):14–23, 2009.

Zheng-Chu Guo and Ding-Xuan Zhou. Concentration estimates for learning with unbounded sampling. *Advances in Computational Mathematics*, 38(1):207–223, 2013.

Frank R. Hampel, Elvezio M. Ronchetti, Peter J. Rousseeuw, and Werner A. Stahel. *Robust Statistics: The Approach Based on Influence Functions*. John Wiley & Sons, New York, 1986.

Ran He, Wei-Shi Zheng, and Bao-Gang Hu. Maximum correntropy criterion for robust face recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(8): 1561–1576, 2011a.

Ran He, Wei-Shi Zheng, Bao-Gang Hu, and Xiang-Wei Kong. A regularized correntropy framework for robust pattern recognition. *Neural Computation*, 23(8):2074–2100, 2011b.

Paul W. Holland and Roy E. Welsch. Robust regression using iteratively reweighted least-squares. *Communications in Statistics-Theory and Methods*, 6(9):813–827, 1977.

Ting Hu, Jun Fan, Qiang Wu, and Ding-Xuan Zhou. Learning theory approach to minimum error entropy criterion. *Journal of Machine Learning Research*, 14(1):377–397, 2013.

Ting Hu, Jun Fan, Qiang Wu, and Ding-Xuan Zhou. Regularization schemes for minimum error entropy principle. *Analysis and Applications*, 2014.

Peter J. Huber. *Robust Statistics*. John Wiley & Sons, New York, 1981.

Weifeng Liu, Puskal P. Pokharel, and José C. Príncipe. Correntropy: properties and applications in non-gaussian signal processing. *Signal Processing, IEEE Transactions on*, 55 (11):5286–5298, 2007.

Shahar Mendelson and Joseph Neeman. Regularization in kernel learning. *Annals of Statistics*, 38(1):526–565, 2010.

José C. Príncipe. *Information Theoretic Learning: Rényi's Entropy and Kernel Perspectives*. Springer, New York, 2010.

Ignacio Santamaría, Puskal P. Pokharel, and José C. Príncipe. Generalized correlation function: definition, properties, and application to blind equalization. *Signal Processing, IEEE Transactions on*, 54(6):2187–2197, 2006.

Bernhard Schölkopf, Alex J. Smola, Robert C. Williamson, and Peter L. Bartlett. New support vector algorithms. *Neural Computation*, 12(5):1207–1245, 2000.

Steve Smale and Ding-Xuan Zhou. Estimating the approximation error in learning theory. *Analysis and Applications*, 1(01):17–41, 2003.

Alex J. Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, 2004.

Ingo Steinwart and Andreas Christmann. *Support Vector Machines*. Springer, New York, 2008.

Ingo Steinwart, Don Hush, and Clint Scovel. Optimal rates for regularized least squares regression. In *Proceedings of the 22nd Conference on Learning Theory*, 2009.

Johan A. K. Suykens, Jos De Brabanter, Lukas Lukas, and Joos Vandewalle. Weighted least squares support vector machines: robustness and sparse approximation. *Neurocomputing*, 48(1):85–105, 2002a.

Johan A. K. Suykens, Tony Van Gestel, Jos De Brabanter, Bart De Moor, and Joos Vandewalle. *Least Squares Support Vector Machines*. World Scientific, Singapore, 2002b.

Michael E. Tipping. Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244, 2001.

Vladimir Vapnik. *Statistical Learning Theory*. John Wiley & Sons, New York, 1998.

Cheng Wang and Ding-Xuan Zhou. Optimal learning rates for least squares regularized regression with unbounded sampling. *Journal of Complexity*, 27(1):55–67, 2011.

Xueqin Wang, Yunlu Jiang, Mian Huang, and Heping Zhang. Robust variable selection with exponential squared loss. *Journal of the American Statistical Association*, 108(502): 632–643, 2013.

Qiang Wu, Yiming Ying, and Ding-Xuan Zhou. Learning rates of least-square regularized regression. *Foundations of Computational Mathematics*, 6(2):171–192, 2006.

Qiang Wu, Yiming Ying, and Ding-Xuan Zhou. Multi-kernel regularized classifiers. *Journal of Complexity*, 23(1):108–134, 2007.

Ding-Xuan Zhou. The covering number in learning theory. *Journal of Complexity*, 18(3): 739–767, 2002.

Journal of Machine Learning Research 16 (2015) 1035-1062 Submitted 10/13; Revised 9/14; Published

Joint Estimation of Multiple Precision Matrices with Common Structures

Wonyul Lee WONYULL@EMAIL.UNC.EDU
Department of Statistics and Operations Research
University of North Carolina
Chapel Hill, NC 27599-3260, USA

Yufeng Liu YFLIU@EMAIL.UNC.EDU
Department of Statistics and Operations Research
Department of Genetics
Department of Biostatistics
Carolina Center for Genome Sciences
University of North Carolina
Chapel Hill, NC 27599-3260, USA

Editor: Francis Bach

Abstract

Estimation of inverse covariance matrices, known as precision matrices, is important in various areas of statistical analysis. In this article, we consider estimation of multiple precision matrices sharing some common structures. In this setting, estimating each precision matrix separately can be suboptimal as it ignores potential common structures. This article proposes a new approach to parameterize each precision matrix as a sum of common and unique components and estimate multiple precision matrices in a constrained l_1 minimization framework. We establish both estimation and selection consistency of the proposed estimator in the high dimensional setting. The proposed estimator achieves a faster convergence rate for the common structure in certain cases. Our numerical examples demonstrate that our new estimator can perform better than several existing methods in terms of the entropy loss and Frobenius loss. An application to a glioblastoma cancer data set reveals some interesting gene networks across multiple cancer subtypes.

Keywords: covariance matrix, graphical model, high dimension, joint estimation, precision matrix

1. Introduction

Estimation of a precision matrix, which is an inverse covariance matrix, has attracted a lot of attention recently. One reason is that the precision matrix plays an important role in various areas of statistical analysis. For example, some classification techniques such as linear discriminant analysis and quadratic discriminant analysis require good estimates of precision matrices. In addition, estimation of a precision matrix is essential to establish conditional dependence relationships in the context of Gaussian graphical models. Another reason is that the high-dimensional nature of many modern statistical applications makes the problem of estimating a precision matrix very challenging. In situations where the

dimension p is comparable to or much larger than the sample size n, more feasible and stable techniques are required for accurate estimation of a precision matrix.

To tackle such problems, various penalized maximum likelihood methods have been considered by many researchers in recent years (Yuan and Lin, 2007; Banerjee et al., 2008; Friedman et al., 2008; Rothman et al., 2008; Lam and Fan, 2009; Fan et al., 2009, and many more). These approaches produce a sparse estimator of the precision matrix by maximizing the penalized Gaussian likelihood with sparse penalties such as the l_1 penalty and the smoothly clipped absolute deviation penalty (Fan and Li, 2001). Ravikumar et al. (2011) studied the theoretical properties of the l_1 penalized likelihood estimator for a broad class of population distributions.

Instead of using likelihood approaches, several techniques take advantage of the connection between linear regression and the entries of the precision matrix. See for example Meinshausen and Bühlmann (2006); Peng et al. (2009); Yuan (2010). In particular, these approaches convert the estimation problem of the precision matrix into relevant regression problems and solve them with sparse regression techniques accordingly. One advantage of these approaches is that they can handle a wide range of distributions including the Gaussian case. Cai et al. (2011) recently proposed a very interesting method to directly estimate the precision matrix without the Gaussian distributional assumption. This approach solves a constrained l_1 minimization problem to obtain a sparse estimator of the precision matrix. They showed that the proposed estimator has a faster convergence rate than the l_1 penalized likelihood estimator for some non-Gaussian cases.

All aforementioned approaches focus on estimation of a single precision matrix. The fundamental assumption of these approaches is that all observations follow the same distribution. However, in some real applications, this assumption can be unreasonable. As a motivating example, consider the glioblastoma multiforme (GBM) cancer data set studied by The Cancer Genome Atlas Research Network (The Cancer Genome Atlas Research Network, 2008). It is shown in the literature that the GBM cancer can be classified into four subtypes (Verhaak et al., 2010). In this case, it would be more realistic to assume that the distribution of gene expression levels can vary from one subtype to another, which results in multiple precision matrices to estimate (Lee et al., 2012). A naive way to estimate them is to model each subtype separately. However, in this separate approach, modeling of one subtype completely ignores the information on other subtypes. This can be suboptimal if there exists some common structure across different subtypes.

To improve the estimation in presence of some common structure, several joint estimation methods have been proposed recently in a penalized likelihood framework. See for example Guo et al. (2011); Honorio and Samaras (2012); Danaher et al. (2014). These methods employ various group penalties in the Gaussian likelihood framework to link the estimation of separate precision matrices.

In this article, we propose a new method to jointly estimate multiple precision matrices. Our approach uses a novel representation of each precision matrix as a sum of common and unique matrices. Then we apply sparse constrained optimization on the common and unique components. The proposed method is applicable for a broad class of distributions including both the Gaussian and some non-Gaussian cases. The main strength of our method is that it uses all available information to jointly estimate the common and unique structures, which can be more preferable than separate modelings. The estimation can be improved

if the precision matrices are similar to each other. Furthermore, our method is able to discover unique structures of each precision matrix, which enables us to identify differences among multiple precision matrices. The proposed estimator is shown to achieve a faster convergence rate for the common structures in certain cases.

The rest of this article is organized as follows. In Section 2, we introduce our proposed method after reviewing some existing separate approaches. We establish its theoretical properties in Section 3. Section 4 develops computational algorithms to obtain a solution for the proposed method. Simulated examples are presented in Section 5 to demonstrate performance of our estimator and analysis of a glioblastoma cancer data example is provided in Section 6. The proofs of theorems are provided in Appendix.

2. Methodology

In this section, we introduce a new method for estimating multiple precision matrices in an l_1 minimization framework. Consider a heterogeneous data set with G different groups. For the gth group ($g = 1, \ldots, G$), let $\{x_1^{(g)}, \ldots, x_{n_g}^{(g)}\}$ be an independent and identically distributed random sample of size n_g, where $x_k^{(g)} = (x_{ki}^{(g)}, \ldots, x_{kp}^{(g)})^{\mathrm{T}}$ is a p-dimensional random vector with the covariance matrix $\Sigma_0^{(g)}$ and precision matrix $\Omega_0^{(g)} := (\Sigma_0^{(g)})^{-1}$. For detailed illustration of our proposed method, we first define some notations similar to Cai et al. (2011). For a matrix $X = (x_{ij}) \in \mathcal{R}^{p\times q}$, we define the elementwise l_1 norm $||X||_1 = \sum_{i=1}^{p}\sum_{j=1}^{q}|x_{ij}|$, the elementwise l_∞ norm $|X|_\infty = \max_{1\le i\le p, 1\le j\le q}|x_{ij}|$ and the matrix l_1 norm $||X||_{L_1} = \max_{1\le j\le q}\sum_{i=1}^{p}|x_{ij}|$. For a vector $x = (x_1, \ldots, x_p)^{\mathrm{T}} \in \mathcal{R}^p$, $|x|_1$ and $|x|_\infty$ denote vector l_1 and l_∞ norms respectively. The notation $X \succ 0$ indicates that X is positive definite. Let I be a $p \times p$ identity matrix. For the gth group, $\hat{\Sigma}^{(g)}$ denotes the sample covariance matrix. Write $\Omega_0^{(g)} = (\omega_{ij,0}^{(g)}); g = 1, \ldots, G$.

Our aim is to estimate the precision matrices, $\Omega_0^{(1)}, \ldots, \Omega_0^{(G)}$. The most naive way to achieve this goal is to estimate each precision matrix separately by taking the inverses of the sample covariance matrices. However, in high dimensional cases, the sample covariance matrices are not only unstable for estimating the covariance matrices, but also not invertible. To estimate the precision matrix in high dimensions, various estimators have been introduced in the literature. For example, various l_1 penalized Gaussian likelihood estimators have been studied intensively in the literature (see for example, Yuan and Lin, 2007; Banerjee et al., 2008; Friedman et al., 2008; Rothman et al., 2008). In this framework, the precision matrices can be estimated by solving the following G optimization problems:

$$\min_{\Omega^{(g)}\succ 0} \mathrm{tr}(\hat{\Sigma}^{(g)}\Omega^{(g)}) - \log\{\det(\Omega^{(g)})\} + \lambda_g \sum_{i\neq j}|w_{ij}^{(g)}|, \ g = 1, \ldots, G, \tag{1}$$

where λ_g is a tuning parameter which controls the degree of the sparsity in the estimated precision matrices. Other sparse penalized Gaussian likelihood estimators have been proposed as well (Lam and Fan, 2009; Fan et al., 2009).

Recently, Cai et al. (2011) proposed an interesting method of constrained l_1 minimization for inverse matrix estimation (CLIME), which can be directly implemented using linear programming. In particular, the CLIME estimator of $\Omega_0^{(g)}$ is the solution of the following

optimization problem:

$$\min ||\Omega^{(g)}||_1 \text{ subject to: } |\hat{\Sigma}^{(g)}\Omega^{(g)} - I|_\infty \leq \lambda_g, \tag{2}$$

where $\hat{\Sigma}^{(g)}$ is the sample covariance matrix and λ_g is a tuning parameter. As the optimization problem in (2) does not require symmetry of the solution, the final CLIME estimator is obtained by symmetrizing the solution of (2). The CLIME estimator does not need the Gaussian distributional assumption. Cai et al. (2011) showed that the convergence rate of the CLIME estimator is faster than that of the l_1 penalized Gaussian likelihood estimator if the underlying true distribution has polynomial-type tails.

To estimate multiple precision matrices, $\Omega_0^{(1)}, \ldots, \Omega_0^{(G)}$, we can build G individual models using the optimization problem (1) or (2). However, these separate approaches can be suboptimal when the precision matrices share some common structure. Several recent papers have proposed joint estimations of multiple precision matrices under the Gaussian distributional assumption to improve estimation. In particular, such an estimator is the solution of

$$\min_{\{\Omega\}} \sum_{g=1}^{G} n_g \left[\text{tr}(\hat{\Sigma}^{(g)}\Omega^{(g)}) - \log\{\det(\Omega^{(g)})\}\right] + P(\{\Omega\}),$$

where n_g is the sample size of the g-th group, $\{\Omega\} = \{\Omega^{(1)}, \ldots, \Omega^{(G)}\}$, and $P(\{\Omega\})$ is a penalty function that encourages similarity across the G estimated precision matrices. For example, Guo et al. (2011) employs a non-convex penalty called *hierarchical group penalty* which has the form, $P(\{\Omega\}) = \lambda \sum_{i \neq j} \left(\sum_{g=1}^{G} |\omega_{ij}^{(g)}|\right)^{1/2}$. Honorio and Samaras (2012) adopts a convex penalty, $P(\{\Omega\}) = \lambda \sum_{i \neq j} |(\omega_{ij}^{(1)}, \ldots, \omega_{ij}^{(G)})|_p$ $(p > 1)$ where $|\cdot|_p$ is the vector l_p norm. To separately control the sparsity level and the extent of similarity, Danaher et al. (2014) considered a *fused lasso penalty*, $P(\{\Omega\}) = \lambda_1 \sum_{g=1}^{G} \sum_{i \neq j} |\omega_{ij}^{(g)}| + \lambda_2 \sum_{g<g'} \sum_{ij} |\omega_{ij}^{(g)} - \omega_{ij}^{(g')}|$. In some simulation settings, they showed that the joint estimation can perform better than separate l_1 penalized normal likelihood estimation. As pointed by Ravikumar et al. (2011), these penalized Gaussian likelihood estimators are applicable even for some mild non-Gaussian data since maximizing a penalized likelihood can be interpreted as minimizing a penalized log-determinant Bregman divergence. However, these approaches were mainly designed for Gaussian data and can be less efficient when the underlying distribution becomes far from Gaussian. In this paper, we propose a new joint method for estimating multiple precision matrices, which is less dependent on the distributional assumption and applicable for both Gaussian and non-Gaussian cases.

In our joint estimation method, we take the multi-task learning perspective and first define the common structure M_0 and the unique structure $R_0^{(g)}$ as

$$M_0 := \frac{1}{G} \sum_{g=1}^{G} \Omega_0^{(g)}, R_0^{(g)} := \Omega_0^{(g)} - M_0; g = 1, \ldots, G.$$

It follows from the definition that $\sum_{g=1}^{G} R_0^{(g)} = 0$, and consequently our representation is identifiable. The idea of decomposing parameters into common and individual structures

was previously considered in the context of supervised multi-tasking learning (Evgeniou and Pontil, 2004). Their aim was to improve prediction performance of supervised multi-tasking learning. Here we focus on better estimation of precision matrices with the common and individual structures. The unique structure is defined to capture different strength of the edges across all classes. In a special case that an element of M_0 is zero, then the corresponding nonzero element in $R_0^{(g)}$ can be interpreted as a unique edge. Thus, the unique structure can address differences in magnitude as well as unique edges. If all precision matrices are very similar, then the unique structures defined above would be close to zero. In this case, it can be natural and advantageous to encourage sparsity among $\{R_0^{(1)}, \ldots, R_0^{(G)}\}$ in the estimation. To estimate the precision matrices consistently in high dimensions, it is also necessary to assume some special structure of M_0 as well. In our work, we also assume that M_0 is sparse. To estimate $\{M_0, R_0^{(1)}, \ldots, R_0^{(G)}\}$, we propose the following constrained l_1 minimization criterion:

$$\min\{||M||_1 + \nu \sum_{g=1}^{G} ||R^{(g)}||_1\}$$

$$\text{s.t } |\frac{1}{G}\sum_{g=1}^{G}\{\hat{\Sigma}^{(g)}(M + R^{(g)}) - I\}|_\infty \leq \lambda_1, |\hat{\Sigma}^{(g)}(M + R^{(g)}) - I|_\infty \leq \lambda_2, \sum_{g=1}^{G} R^{(g)} = 0, \quad (3)$$

where λ_1 and λ_2 are tuning parameters and ν is a prespecified weight. Note that if $\lambda_1 > \lambda_2$, then the second inequality constraints in (3) imply the first inequality constraint. Therefore, we only consider a pair of (λ_1, λ_2) satisfying $\lambda_1 \leq \lambda_2$. The first inequality constraint in (3) reflects how close the final estimators are to the inverses of the sample covariance matrices in an average sense. On the other hand, the second inequality constraint controls an individual level of closeness between the estimators and the sample covariance matrices.

For illustration, consider an extreme case where all the precision matrices are the same. In this case, the unique structures may be negligible and the first inequality constraint in (3) approximately reduces to $|(G^{-1}\sum_{g=1}^{G}\hat{\Sigma}^{(g)})M - I|_\infty \leq \lambda_1$. Therefore, we can pool all the sample covariance matrices to estimate the common structure which is the precision matrix in this case. This would be advantageous than building each model separately. The value of ν in (3) reflects how complex the unique structures of the resulting estimators are. If the resulting estimators are expected to be very similar from each other, then a large value of ν is preferred. In Section 3, ν is set to be G^{-1} or $G^{-1/2}$ for our theoretical results.

Similar to Cai et al. (2011), the solutions in (3) are not symmetric in general. Therefore, the final estimators are obtained after a symmetrization step. Let $\{\hat{M}, \hat{R}^{(1)}, \ldots, \hat{R}^{(G)}\}$ be the solution of (3). Then we define $\hat{\Omega}_1^{(g)} := \hat{M} + \hat{R}^{(g)}; g = 1, \ldots, G$. The final estimator of $\{\Omega_0^{(1)}, \ldots, \Omega_0^{(G)}\}$ is obtained by symmetrizing $\{\hat{\Omega}_1^{(1)}, \ldots, \hat{\Omega}_1^{(G)}\}$ as follows. Let $\hat{\Omega}_1^{(g)} = (\hat{\omega}_{ij,1}^{(g)})$. Our joint estimator of multiple precision matrices (JEMP), $\{\hat{\Omega}^{(1)}, \ldots, \hat{\Omega}^{(G)}\}$, is defined as symmetric matrices, $\{\hat{\Omega}^{(g)} = (\hat{\omega}_{ij}^{(g)}); g = 1, \ldots, G\}$ with

$$\hat{\omega}_{ij}^{(g)} = \hat{\omega}_{ij,1}^{(g)} I\{\sum_{g=1}^{G}|\hat{\omega}_{ij,1}^{(g)}| \leq \sum_{g=1}^{G}|\hat{\omega}_{ji,1}^{(g)}|\} + \hat{\omega}_{ji,1}^{(g)} I\{\sum_{g=1}^{G}|\hat{\omega}_{ij,1}^{(g)}| > \sum_{g=1}^{G}|\hat{\omega}_{ji,1}^{(g)}|\}; g = 1, \ldots, G.$$

Note that the solution $\hat{\Omega}^{(g)}$ is not necessarily positive definite. Although there is no guarantee for the solution to be positive definite, it can be positive definite with high probability.

In our simulation study, we observed that within a reasonable range of tuning parameters, almost all solutions are positive definite. Furthermore, one can perform projection of the estimator to the space of positive definite matrices to ensure positive definitiveness as discussed in Yuan (2010).

As a remark, although we focus on generalizing CLIME for multiple graph estimation in this paper, our proposed common and unique structure approach can also be applied to the graphical lasso estimator under the Gaussian assumption as pointed out by one reviewer. As a future research direction, it would be interesting to investigate how the common and unique structure framework works in the graphical lasso estimator.

3. Theoretical Properties

In this section, we investigate theoretical properties of our proposed joint estimator JEMP. In particular, we first construct the convergence rate of our estimator in the high dimensional setting. Then we show that the convergence rate can be improved for the common structure of the precision matrices in certain cases. Finally, the model selection consistency is shown with an additional thresholding step.

For theoretical properties, we follow the set-up of Cai et al. (2011) and the results therein are also used for our technical derivations. In this section, for simplicity, we assume that $n = n_1 = \cdots = n_G$. We consider the following class of matrices,

$$\mathcal{U} := \{\Omega : \Omega \succ 0, \|\Omega\|_{L_1} \leq C_M\},$$

and assume that $\Omega_0^{(g)} \in \mathcal{U}$ for all $g = 1, \ldots, G$. This assumption requires that the true precision matrices are sparse in terms of the l_1 norm while allowing them to have many small entries. Write $E(x^{(g)}) = (\mu_1^{(g)}, \ldots, \mu_p^{(g)})^{\mathrm{T}}$. We also make the following moment condition on $x^{(g)}$ for our theoretical results.

Condition 1 *There exists some $0 < \eta < 1/4$ such that $E[\exp\{t(x_i^{(g)} - \mu_i^{(g)})^2\}] \leq K < \infty$ for all $|t| \leq \eta$ and all i, g and $G \log p/n \leq \eta$, where K is a bounded constant.*

Condition 1 indicates that the components of $x^{(g)}$ are uniformly sub-Gaussian. This condition is satisfied if $x^{(g)}$ follows a multivariate Gaussian distribution or has uniformly bounded components.

Theorem 1 *Assume Condition 1 holds. Let $\lambda_1 = \lambda_2 = 3C_M C_0 (\log p/n)^{1/2}$, where $C_0 = 2\eta^{-2}(2 + \tau + \eta^{-1}e^2K^2)^2$ and $\tau > 0$. Set $\nu = G^{-1}$. Then*

$$\max_{ij} \left(\frac{1}{G} \sum_{g=1}^{G} |\hat{\omega}_{ij}^{(g)} - \omega_{ij,0}^{(g)}| \right) \leq 6C_M^2 C_0 \left(\frac{\log p}{n} \right)^{1/2},$$

with probability greater than $1 - 4Gp^{-\tau}$.

In an average sense, the convergence rate can be viewed the same as that of the CLIME estimator which is of order $(\log p/n)^{1/2}$. In this theorem, the first inequality constraint in (3) does not play any role in the estimation procedure as we set $\lambda_1 = \lambda_2$. In the next theorem, with properly chosen λ_1, we construct a faster convergence rate for the common part under certain conditions.

Theorem 2 *Assume Condition 1 holds. Suppose that there exists* $C_R > 0$ *such that* $\|R_0^{(g)}\|_{L_1} \leq C_R$ *for all* $g = 1, \ldots, G$ *and* $(\sum_{g=1}^{G} \|R_0^{(g)}\|_{L_1}) \leq C_R G^{1/2}$. *Set* $\nu = G^{-1/2}$ *and let* $\lambda_1 = (C_M + C_R)C_0\{\log p/(nG)\}^{1/2}$ *and* $\lambda_2 = C_M C_0(\log p/n)^{1/2}$. *Then*

$$|\hat{M} - M_0|_\infty \leq C_0(2C_M^2 + 4C_M C_R + C_R^2)\left(\frac{\log p}{nG}\right)^{1/2},$$

with probability greater than $1 - 2(1 + 3G)p^{-\tau}$.

Theorem 2 states that our proposed method can estimate the common part more efficiently with the corresponding convergence rate of order $\{\log p/(nG)\}^{1/2}$, which is faster than the order $(\log p/n)^{1/2}$.

Note that our theorems show consistency of our estimator in terms of the elementwise l_∞ norm. On the other hand, Guo et al. (2011) showed consistency of their estimator under the Frobenious norm. Therefore, our theoretical results are not directly comparable to the theorems in Guo et al. (2011). However, it is worthwhile to note that our Theorem 2 reveals the effect of G on the consistency while the theorems in Guo et al. (2011) do not show explicitly how their estimator can have advantage over separate estimation in terms of consistency.

Besides its estimation consistency, we also prove the model selection consistency of our estimator which means that it reveals the exact set of nonzero components in the true precision matrices with high probability. For this result, a thresholding step is introduced. In particular, a threshold estimator $\tilde{\Omega}^{(g)} = (\tilde{\omega}_{ij}^{(g)})$ based on $\{\hat{\Omega}^{(1)}, \ldots, \hat{\Omega}^{(G)}\}$ is defined as,

$$\tilde{\omega}_{ij}^{(g)} = \hat{\omega}_{ij}^{(g)} I\{|\hat{\omega}_{ij}^{(g)}| \geq \delta_n\},$$

where $\delta_n \geq 2C_M G\lambda_2$ and λ_2 is given in Theorem 1. To state the model selection consistency precisely, we define

$$\mathcal{S}_0 := \{(i,j,g) : \omega_{ij,0}^{(g)} \neq 0\}, \hat{\mathcal{S}} := \{(i,j,g) : \tilde{\omega}_{ij}^{(g)} \neq 0\} \text{ and } \theta_{\min} := \min_{(i,j,g)\in\mathcal{S}_0} \sum_{g=1}^{G} |\omega_{ij,0}^{(g)}|.$$

Then the next theorem states the model selection consistency of our estimator.

Theorem 3 *Assume Condition 1 holds. If* $\theta_{\min} > 2\delta_n$, *then*

$$pr(\mathcal{S}_0 = \hat{\mathcal{S}}) \geq 1 - 4Gp^{-\tau}.$$

4. Numerical Algorithm

In this section, we describe how to obtain the numerical solutions of the optimization problem (3). In Section 4.1, the optimization problem (3) is decomposed into p individual subproblems and a linear programming approach is used to solve them. In Section 4.2, we describe another algorithm using the alternating directions method of multiplier (ADMM). Section 4.3 explains how the tuning parameters can be selected.

4.1 Decomposition of (3)

Similar to the Lemma 1 in Cai et al. (2011), one can show that the optimization problem (3) can be decomposed into p individual minimization problems. In particular, let e_i be the ith column of I. For $1 \leq i \leq p$, let $\{\hat{m}_i, \hat{r}_i^{(1)}, \ldots, \hat{r}_i^{(\mathrm{G})}\}$ be the solution of the following optimization problem:

$$\min\{|m|_1 + \nu \sum_{g=1}^{G} |r^{(\mathrm{g})}|_1\}$$

$$\text{s.t. } |\frac{1}{G}\sum_{g=1}^{G}\{\hat{\Sigma}^{(\mathrm{g})}(m + r^{(\mathrm{g})}) - e_i\}|_\infty \leq \lambda_1, |\hat{\Sigma}^{(\mathrm{g})}(m + r^{(\mathrm{g})}) - e_i|_\infty \leq \lambda_2, \sum_{g=1}^{G} r^{(\mathrm{g})} = 0, \qquad (4)$$

where m, $r^{(1)}, \ldots, r^{(\mathrm{G})}$ are vectors in $\mathcal{R}^p$. We can show that solving the optimization problem (3) is equivalent to solving the p optimization problems in (4). The optimization problem in (4) can be further reformulated as a linear programming problem and the simplex method is used to solve this problem (Boyd and Vandenberghe, 2004). For our simulation study and the GBM data analysis, we obtain the solution of (3) using the efficient R-package *fastclime*, which provides a generic fast linear programming solver (Pang et al., 2014).

4.2 An ADMM Algorithm

In this section, we describe an alternating directions method of multipliers (ADMM) algorithm to solve (4) which can be potentially more scalable than the previously explained linear programming approach. We refer the reader to Boyd et al. (2010) for detailed explanation of ADMM algorithms and their convergence properties.

To reformulate (4) into an appropriate ADMM form, define $y = (m^{\mathrm{T}}, \nu r^{(1)^{\mathrm{T}}}, \ldots, \nu r^{(\mathrm{G})^{\mathrm{T}}})^{\mathrm{T}}$, $z_m = \sum_{g=1}^{G}\{\hat{\Sigma}^{(\mathrm{g})}(m + r^{(\mathrm{g})}) - e_i\}/G$, $z_g = \hat{\Sigma}^{(\mathrm{g})}(m + r^{(\mathrm{g})}) - e_i$, and $z = ({z_1}^{\mathrm{T}}, \ldots, {z_G}^{\mathrm{T}}, {z_m}^{\mathrm{T}})^{\mathrm{T}}$. Denote the $a \times a$ identity matrix as $I_{a\times a}$ and the $a \times b$ zero matrix as $O_{a\times b}$. Then the problem (4) can be rewritten as

$$\min |y|_1 \text{ s.t. } |z_m|_\infty \leq \lambda_1, |z_g|_\infty \leq \lambda_2, Ay - Bz = C, \text{ where} \qquad (5)$$

$$A = \begin{pmatrix} \hat{\Sigma}^{(1)} & \nu^{-1}\hat{\Sigma}^{(1)} & O_{p\times p} & \cdots & O_{p\times p} \\ \hat{\Sigma}^{(2)} & O_{p\times p} & \nu^{-1}\hat{\Sigma}^{(2)} & \cdots & O_{p\times p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \hat{\Sigma}^{(\mathrm{G})} & O_{p\times p} & O_{p\times p} & \cdots & \nu^{-1}\hat{\Sigma}^{(\mathrm{G})} \\ G^{-1}\sum_{g=1}^{G}\hat{\Sigma}^{(\mathrm{g})} & (\nu G)^{-1}\hat{\Sigma}^{(1)} & (\nu G)^{-1}\hat{\Sigma}^{(2)} & \cdots & (\nu G)^{-1}\hat{\Sigma}^{(\mathrm{G})} \\ O_{p\times p} & I_{p\times p} & I_{p\times p} & \cdots & I_{p\times p} \end{pmatrix},$$

$B = \begin{pmatrix} I_{(1+G)p\times(1+G)p} \\ O_{p\times(1+G)p} \end{pmatrix}$, and $C = ({e_i}^{\mathrm{T}}, \ldots, {e_i}^{\mathrm{T}}, O_{p\times 1})^{\mathrm{T}}$. The scaled augmented Lagrangian for (5) is given by

$$L(y, z, u) = |y|_1 + \frac{\rho}{2}||Ay - Bz - C + u||_2^2, \text{ s.t. } |z_m|_\infty \leq \lambda_1, |z_g|_\infty \leq \lambda_2,$$

where u is a $(2+G)p$-dimensional vector of dual variables. With the current solution z^k, u^k, the ADMM algorithm updates solutions sequentially as follows:

(a) $y^{k+1} = \text{argmin}_y L(y, z^k, u^k)$.

(b) $z^{k+1} = \text{argmin}_z L(y^{k+1}, z, u^k)$, s.t. $|z_m|_\infty \leq \lambda_1, |z_g|_\infty \leq \lambda_2$.

(c) $u^{k+1} = u^k + Ay^{k+1} - Bz^{k+1} - c$.

As $\text{argmin}_y L(y, z^k, u^k) = \text{argmin}_y\{|y|_1 + \frac{\rho}{2}||Ay - Bz^k - C + u^k||_2^2\}$, the step (a) can be viewed as an L_1 penalized least squares problem. Therefore, the step (a) can be solved using some existing algorithms for L_1 penalized least squares problems. In addition, one can show that the step (b) has a closed form of solution, $z^{k+1} = \min\{\max\{A'y^{k+1} - C' + (u^k)', -\lambda\}, \lambda\}$ where A' is the submatrix of A consisting of the first $(1+G)p$ rows, C' and $(u^k)'$ are the corresponding subvectors of C and u^k, and λ is a $(1+G)p$-dimensional vector of which the first Gp elements are λ_2 and the rest are λ_1. Note that scalability and computational speed of this ADMM algorithm largely depend on the algorithm used for the step (a) as the other steps have the explicit form of solutions.

4.3 Tuning Parameter Selection

To apply our method, we need to choose the tuning parameters, λ_1 and λ_2. In practice, we construct several models with many pairs of λ_1 and λ_2 satisfying $\lambda_1 \leq \lambda_2$ and evaluate them to determine the optimal pair. To evaluate each estimator, we measure the likelihood loss (LL) used in Cai et al. (2011) and its definition is

$$\text{LL} = \sum_{g=1}^{G} \text{tr}(\hat{\Sigma}_v^{(\text{g})} \hat{\Omega}^{(\text{g})}) - \log\{\det(\hat{\Omega}^{(\text{g})})\},$$

where $\hat{\Sigma}_v^{(\text{g})}$ is the sample covariance matrix of the gth group computed from an independent validation set. As mentioned in Section 2, the likelihood loss can be applicable for both Gaussian and some non-Gaussian data as it corresponds to the log-determinant Bregman divergence between the estimators and empirical precision matrices in the validation set. Among several pairs of tuning values, we select the pair which minimizes LL. If a validation set is not available, a K-fold cross-validation can be combined to this criterion. In particular, we first randomly split the data set into K parts of equal sizes. Denote the data in the kth part by $\{X_{(k)}^{(1)}, \ldots, X_{(k)}^{(\text{G})}\}$ which is used as a validation set for the kth estimator. For each k, with a given value of (λ_1, λ_2), we obtain estimators using all observations which do not belong to $\{X_{(k)}^{(1)}, \ldots, X_{(k)}^{(\text{G})}\}$ and denote them as $\{\hat{\Omega}_{(k)}^{(\text{G})}, \ldots, \hat{\Omega}_{(k)}^{(\text{G})}\}$. Then the likelihood loss (LL) is defined as

$$\text{LL} = \sum_{k=1}^{K} \sum_{g=1}^{G} \text{tr}(\hat{\Sigma}_{(k)}^{(\text{g})} \hat{\Omega}_{(k)}^{(\text{g})}) - \log\{\det(\hat{\Omega}_{(k)}^{(\text{g})})\},$$

where $\hat{\Sigma}_{(k)}^{(\text{g})}$ is the sample covariance matrix of the gth group using $X_{(k)}^{(\text{g})}$. Once the optimal pair is selected which minimizes LL, the final model is constructed using all data points with the selected pair.

5. Simulated Examples

In this section, we carry out simulation studies to assess the numerical performance of our proposed method. In particular, we compare the numerical performance of five methods: two separate methods and three joint methods. In separate approaches, each precision matrix is estimated separately via the CLIME estimator or the GLASSO estimator. For joint approaches, all precision matrices are estimated together using our JEMP estimator, the fused graphical lasso (FGL) estimator by Danaher et al. (2014), or the estimator by Guo et al. (2011), which we refer to as JOINT estimator hereafter. In our proposed method, ν is set to be $G^{-1/2}$. We also tried different values of ν such as G^{-1}, and the results are similar thus omitted. We consider three models as described below: the first two from Guo et al. (2011) and the last from Rothman et al. (2008); Cai et al. (2011). In all models, we set $p = 100$, $G = 3$ and $\Omega_0^{(g)} = \Omega_c + U^{(g)}$, where Ω_c is common in all groups and $U^{(g)}$ represents unique structure to the gth group. The common part, Ω_c, is generated as follows:

Model 1. Ω_c is a tridiagonal precision matrix. In particular, $\Sigma_c := \Omega_c^{-1} = (\sigma_{ij})$ is first constructed, where $\sigma_{ij} = \exp(-|d_i - d_j|/2)$, $d_1 < \ldots < d_p$, and $d_i - d_{i-1} \sim \text{Unif}(0.5, 1), i = 2, \ldots, p$. Then let $\Omega_c = \Sigma_c^{-1}$.

Model 2. Ω_c is a 3 nearest-neighbor network. In particular, p points are randomly picked on a unit square and all pairwise distances among the points are calculated. Then we find 3 nearest neighbors for each point and a pair of symmetric entries in Ω_c corresponding to a pair of neighbors has a value randomly chosen from the interval $[-1, -0.5] \cup [0.5, 1]$.

Model 3. $\Omega_c = \Gamma + \delta I$, where each off-diagonal entry in Γ is generated independently from $0.5y$, with y following the Bernoulli distribution with success probability 0.02. Here, δ is selected so that the condition number of Ω_c is equal to p.

For each $U^{(g)}$, we randomly pick a pair of symmetric off-diagonal entries and replace them with values randomly chosen from the interval $[-1, -0.5] \cup [0.5, 1]$. We repeat this procedure until $\sum_{i<j} I(|u_{ij}^{(g)}| > 0) / \sum_{i<j} I(|\omega_{ij,c}| > 0) = \rho$, where $\Omega_c = (\omega_{ij,c})$ and $U^{(g)} = u_{ij}^{(g)}$. Therefore, ρ is the ratio of the number of unique nonzero entries to the number of common nonzero entries. We consider four values of $\rho = 0, 0.25, 1$ and 4. To make the resulting precision matrices positive-definite, each diagonal element of each matrix $\Omega_0^{(g)}$ is replaced with 1.5 times the sum of the absolute values of the corresponding row. Finally, each matrix $\Omega_0^{(g)}$ is standardized to have unit diagonals. Note that in the case of $\rho = 1$ or 4, the true precision matrices are quite different from each other. From these cases, we can assess how joint methods work when the precision matrices are not similar. In addition, we also consider Model 4 below to assess how JEMP works when the precision matrices have different structures from each other.

Model 4. $\Omega_0^{(1)}$ is the tridiagonal precision matrix as in Model 1, $\Omega_0^{(2)}$ is the 3 nearest-neighbor network in Model 2, and $\Omega_0^{(3)}$ is the random network in Model 3.

For each group in each model, we generate a training sample of size $n = 100$ from either a multivariate normal distribution $N(0, \Sigma_0^{(g)})$ or a multivariate t-distribution with the covariance matrix $\Sigma_0^{(g)}$ and degrees of freedom of 3 or 5. In order to select optimal tuning parameters, an independent validation set of size $n = 100$ is also generated from the same distribution of the training sample. For each estimator, optimal tuning parameters are selected as described in Section 4. We replicate simulations 50 times for each model.

		$\rho = 0$		$\rho = 0.25$	
		EL	FL	EL	FL
Normal	CLIME	4.42 (0.02)	8.57 (0.03)	4.35 (0.02)	8.42 (0.03)
	GLASSO	3.70 (0.02)	6.90 (0.03)	3.60 (0.02)	6.73 (0.03)
	JOINT	3.43 (0.02)	6.64 (0.04)	3.41 (0.02)	6.61 (0.03)
	FGL	1.99 (0.02)	3.75 (0.03)	2.09 (0.02)	3.92 (0.03)
	JEMP	2.08 (0.02)	4.06 (0.04)	2.20 (0.02)	4.31 (0.04)
t (DF=5)	CLIME	5.75 (0.17)	10.63 (0.26)	5.81 (0.19)	10.75 (0.33)
	GLASSO	5.60 (0.09)	10.23 (0.16)	5.45 (0.09)	10.00 (0.16)
	JOINT	5.08 (0.11)	9.44 (0.15)	5.01 (0.12)	9.28 (0.19)
	FGL	3.47 (0.07)	6.12 (0.11)	3.46 (0.08)	6.12 (0.11)
	JEMP	3.21 (0.06)	6.14 (0.11)	3.41 (0.10)	6.52 (0.19)
t (DF=3)	CLIME	10.34 (0.83)	18.08 (1.05)	10.15 (0.91)	17.25 (1.06)
	GLASSO	11.87 (0.33)	24.10 (0.95)	11.78 (0.33)	24.21 (0.95)
	JOINT	8.84 (0.58)	15.16 (0.85)	8.95 (0.66)	15.17 (0.92)
	FGL	7.01 (0.24)	12.39 (0.52)	7.40 (0.31)	13.23 (0.66)
	JEMP	6.02 (0.33)	11.56 (0.73)	5.95 (0.30)	11.16 (0.62)

		$\rho = 1$		$\rho = 4$	
		EL	FL	EL	FL
Normal	CLIME	4.23 (0.02)	8.15 (0.03)	3.67 (0.01)	6.95 (0.03)
	GLASSO	3.37 (0.02)	6.33 (0.03)	2.57 (0.01)	4.96 (0.03)
	JOINT	3.27 (0.01)	6.40 (0.03)	2.51 (0.01)	4.95 (0.02)
	FGL	2.18 (0.01)	4.07 (0.02)	1.82 (0.01)	3.47 (0.02)
	JEMP	2.38 (0.01)	4.77 (0.04)	2.11 (0.01)	4.28 (0.02)
t (DF=5)	CLIME	5.53 (0.16)	10.12 (0.23)	4.83 (0.17)	8.72 (0.25)
	GLASSO	5.11 (0.09)	9.54 (0.17)	4.28 (0.09)	8.35 (0.19)
	JOINT	4.71 (0.10)	8.71 (0.14)	3.87 (0.12)	7.03 (0.16)
	FGL	3.31 (0.07)	5.95 (0.11)	2.54 (0.06)	4.68 (0.10)
	JEMP	3.32 (0.07)	6.40 (0.13)	2.78 (0.07)	5.35 (0.12)
t (DF=3)	CLIME	9.89 (0.86)	17.82 (1.16)	8.93 (0.91)	16.58 (1.28)
	GLASSO	11.32 (0.32)	23.77 (0.99)	10.42 (0.31)	23.70 (1.05)
	JOINT	9.27 (1.68)	14.23 (1.26)	7.14 (0.65)	11.90 (0.72)
	FGL	6.51 (0.25)	11.73 (0.56)	5.95 (0.27)	11.55 (0.67)
	JEMP	5.71 (0.29)	10.99 (0.73)	4.72 (0.24)	9.04 (0.49)

Table 1: Comparison summaries using Entropy loss (EL) and Frobenius loss (FL) over 50 replications for Model 1.

		$\rho = 0$		$\rho = 0.25$	
		EL	FL	EL	FL
Normal	CLIME	5.10 (0.02)	9.80 (0.04)	5.05 (0.02)	9.68 (0.04)
	GLASSO	4.50 (0.02)	8.07 (0.03)	4.44 (0.02)	7.98 (0.03)
	JOINT	3.89 (0.02)	7.42 (0.04)	4.13 (0.02)	7.84 (0.04)
	FGL	2.26 (0.02)	4.26 (0.03)	2.70 (0.02)	5.02 (0.03)
	JEMP	2.31 (0.02)	4.44 (0.03)	2.80 (0.02)	5.36 (0.03)
t (DF=5)	CLIME	6.60 (0.17)	12.03 (0.25)	6.62 (0.19)	12.09 (0.32)
	GLASSO	6.78 (0.09)	11.67 (0.15)	6.56 (0.09)	11.37 (0.14)
	JOINT	6.16 (0.10)	11.18 (0.16)	6.12 (0.14)	11.14 (0.23)
	FGL	4.03 (0.07)	6.88 (0.11)	4.28 (0.07)	7.30 (0.10)
	JEMP	3.74 (0.06)	6.98 (0.11)	4.15 (0.09)	7.72 (0.20)
t (DF=3)	CLIME	11.41 (0.87)	19.55 (1.06)	11.16 (0.93)	18.66 (1.09)
	GLASSO	13.16 (0.34)	24.31 (0.88)	12.90 (0.34)	24.29 (0.88)
	JOINT	10.14 (0.56)	16.96 (0.80)	10.24 (0.68)	17.03 (0.94)
	FGL	8.34 (0.28)	13.78 (0.55)	8.55 (0.31)	14.16 (0.59)
	JEMP	7.17 (0.36)	13.31 (0.84)	7.08 (0.31)	12.76 (0.61)

		$\rho = 1$		$\rho = 4$	
		EL	FL	EL	FL
Normal	CLIME	4.84 (0.02)	9.27 (0.04)	3.77 (0.01)	7.14 (0.03)
	GLASSO	4.07 (0.02)	7.42 (0.03)	2.68 (0.01)	5.09 (0.02)
	JOINT	3.99 (0.01)	7.72 (0.03)	2.63 (0.01)	5.16 (0.02)
	FGL	2.99 (0.01)	5.51 (0.02)	1.98 (0.01)	3.74 (0.01)
	JEMP	3.20 (0.01)	6.34 (0.04)	2.35 (0.01)	4.74 (0.02)
t (DF=5)	CLIME	6.14 (0.16)	11.22 (0.24)	4.95 (0.17)	8.96 (0.25)
	GLASSO	5.85 (0.09)	10.52 (0.16)	4.44 (0.09)	8.56 (0.18)
	JOINT	5.44 (0.10)	10.05 (0.15)	4.02 (0.12)	7.32 (0.16)
	FGL	4.07 (0.07)	7.17 (0.10)	2.68 (0.06)	4.91 (0.10)
	JEMP	4.11 (0.06)	7.87 (0.13)	3.00 (0.07)	5.77 (0.13)
t (DF=3)	CLIME	10.53 (0.88)	18.53 (1.15)	9.10 (0.92)	16.84 (1.29)
	GLASSO	12.11 (0.32)	23.89 (0.93)	10.59 (0.32)	23.77 (1.04)
	JOINT	10.00 (1.67)	15.26 (1.26)	7.27 (0.64)	12.10 (0.72)
	FGL	7.23 (0.25)	12.34 (0.52)	6.02 (0.26)	11.50 (0.64)
	JEMP	6.59 (0.31)	12.19 (0.70)	4.99 (0.26)	9.48 (0.53)

Table 2: Comparison summaries using Entropy loss (EL) and Frobenius loss (FL) over 50 replications for Model 2.

		$\rho = 0$		$\rho = 0.25$	
		EL	FL	EL	FL
Normal	CLIME	3.62 (0.02)	6.87 (0.03)	3.92 (0.02)	7.51 (0.04)
	GLASSO	2.60 (0.01)	5.03 (0.03)	3.03 (0.01)	5.78 (0.03)
	JOINT	2.53 (0.01)	4.97 (0.02)	2.99 (0.01)	5.89 (0.03)
	FGL	1.54 (0.01)	2.95 (0.02)	2.21 (0.01)	4.16 (0.02)
	JEMP	1.80 (0.01)	3.61 (0.03)	2.48 (0.01)	4.96 (0.03)
t (DF=5)	CLIME	4.77 (0.17)	8.68 (0.26)	5.23 (0.19)	9.63 (0.33)
	GLASSO	4.32 (0.09)	8.42 (0.20)	4.82 (0.09)	9.11 (0.18)
	JOINT	3.84 (0.12)	7.02 (0.16)	4.43 (0.15)	8.10 (0.21)
	FGL	2.54 (0.06)	4.68 (0.10)	3.11 (0.07)	5.62 (0.10)
	JEMP	2.60 (0.06)	4.99 (0.11)	3.35 (0.10)	6.44 (0.18)
t (DF=3)	CLIME	9.08 (0.84)	16.05 (1.07)	9.40 (0.92)	15.92 (1.06)
	GLASSO	10.64 (0.33)	24.09 (1.06)	11.14 (0.33)	24.26 (1.01)
	JOINT	7.54 (0.57)	13.03 (0.87)	8.35 (0.66)	14.09 (0.89)
	FGL	5.87 (0.26)	11.39 (0.65)	6.72 (0.30)	12.53 (0.70)
	JEMP	5.05 (0.37)	10.10 (0.93)	5.49 (0.30)	10.44 (0.66)

		$\rho = 1$		$\rho = 4$	
		EL	FL	EL	FL
Normal	CLIME	4.33 (0.02)	8.33 (0.03)	4.03 (0.02)	7.68 (0.03)
	GLASSO	3.52 (0.02)	6.54 (0.03)	3.00 (0.01)	5.67 (0.03)
	JOINT	3.50 (0.01)	6.86 (0.02)	2.94 (0.01)	5.78 (0.02)
	FGL	2.90 (0.01)	5.37 (0.02)	2.28 (0.01)	4.28 (0.01)
	JEMP	3.17 (0.01)	6.40 (0.02)	2.66 (0.01)	5.40 (0.02)
t (DF=5)	CLIME	5.64 (0.16)	10.31 (0.23)	5.20 (0.17)	9.42 (0.26)
	GLASSO	5.31 (0.09)	9.81 (0.17)	4.71 (0.09)	8.93 (0.18)
	JOINT	4.91 (0.11)	9.09 (0.14)	4.29 (0.12)	7.86 (0.17)
	FGL	3.66 (0.06)	6.53 (0.10)	2.98 (0.07)	5.40 (0.10)
	JEMP	3.93 (0.07)	7.56 (0.12)	3.27 (0.07)	6.32 (0.14)
t (DF=3)	CLIME	10.00 (0.87)	17.87 (1.16)	9.36 (0.88)	17.25 (1.26)
	GLASSO	11.60 (0.32)	23.89 (0.97)	10.89 (0.31)	23.79 (0.99)
	JOINT	9.52 (1.68)	14.60 (1.27)	7.57 (0.63)	12.59 (0.71)
	FGL	6.71 (0.24)	11.84 (0.52)	6.36 (0.26)	11.87 (0.61)
	JEMP	5.90 (0.26)	11.02 (0.59)	5.20 (0.26)	9.70 (0.51)

Table 3: Comparison summaries using Entropy loss (EL) and Frobenius loss (FL) over 50 replications for Model 3.

	Normal		t (DF=5)		t (DF=3)	
	EL	FL	EL	FL	EL	FL
CLIME	4.39 (0.02)	8.45 (0.04)	6.06 (0.39)	10.82 (0.43)	10.59 (1.03)	17.35 (1.08)
GLASSO	3.62 (0.02)	6.71 (0.03)	5.57 (0.11)	10.02 (0.14)	11.79 (0.43)	24.06 (1.29)
JOINT	3.68 (0.01)	7.16 (0.03)	5.24 (0.14)	9.56 (0.17)	8.28 (0.37)	13.83 (0.50)
FGL	3.12 (0.01)	5.75 (0.02)	3.85 (0.07)	6.84 (0.11)	7.08 (0.33)	12.26 (0.71)
JEMP	3.50 (0.01)	7.04 (0.02)	4.27 (0.08)	8.17 (0.14)	6.22 (0.29)	11.27 (0.60)

Table 4: Comparison summaries using Entropy loss (EL) and Frobenius loss (FL) over 50 replications for Model 4.

To compare performance of five different methods, we use the average entropy loss and the average Frobenius loss defined as,

$$\text{EL} = G^{-1}\sum_{g=1}^{G}\left\{\text{tr}(\Sigma_0^{(g)}\hat{\Omega}^{(g)}) - \log\det(\Sigma_0^{(g)}\hat{\Omega}^{(g)}) - p\right\},$$

$$\text{FL} = G^{-1}\sum_{g=1}^{G}\| \Omega_0^{(g)} - \hat{\Omega}^{(g)} \|_F^2,$$

where $\| \, . \, \|_F$ is the Frobenius norm of a matrix.

Table 1 reports the results for Model 1. In terms of estimation accuracy, the three joint estimation methods, JEMP, FGL, and JOINT, outperform the two separate estimation methods while JEMP and FGL show better performance than JOINT. In Gaussian cases, FGL exhibits slightly smaller losses than JEMP. However, JEMP outperforms FGL in terms of entropy loss for some cases when the underlying distribution is t_5. If the true underlying distribution is t_3, then JEMP clearly outperforms FGL in both entropy loss and Frobenius loss for all cases. This indicates that our proposed JEMP can have some advantage in estimation for some non-Gaussian data. Overall, JEMP shows very competitive performance compared with other methods. Tables 2-3 report the results for Models 2 and 3 respectively. Performances of the methods show similar patterns as in Model 1. JEMP and FGL perform best while FGL is slightly better in Gaussian cases and JEMP has the best performance in the t_3 case.

Table 4 summarizes the results for Model 4 in which the true precision matrices have different structures. As in Models 1-3, our method outperforms JOINT, CLIME, and GLASSO for all cases. It shows competitive performance with FGL when the distribution is Gaussian or t_5. However, it outperforms FGL in the case of t_3 distribution. This indicates that our method works as well even when structures of precision matrices are different from each other. Note that the precision matrices in Model 4 share many zero components although their main structures are different. Joint methods can work better here since they encourage many common zeros to be estimated as zeros simultaneously.

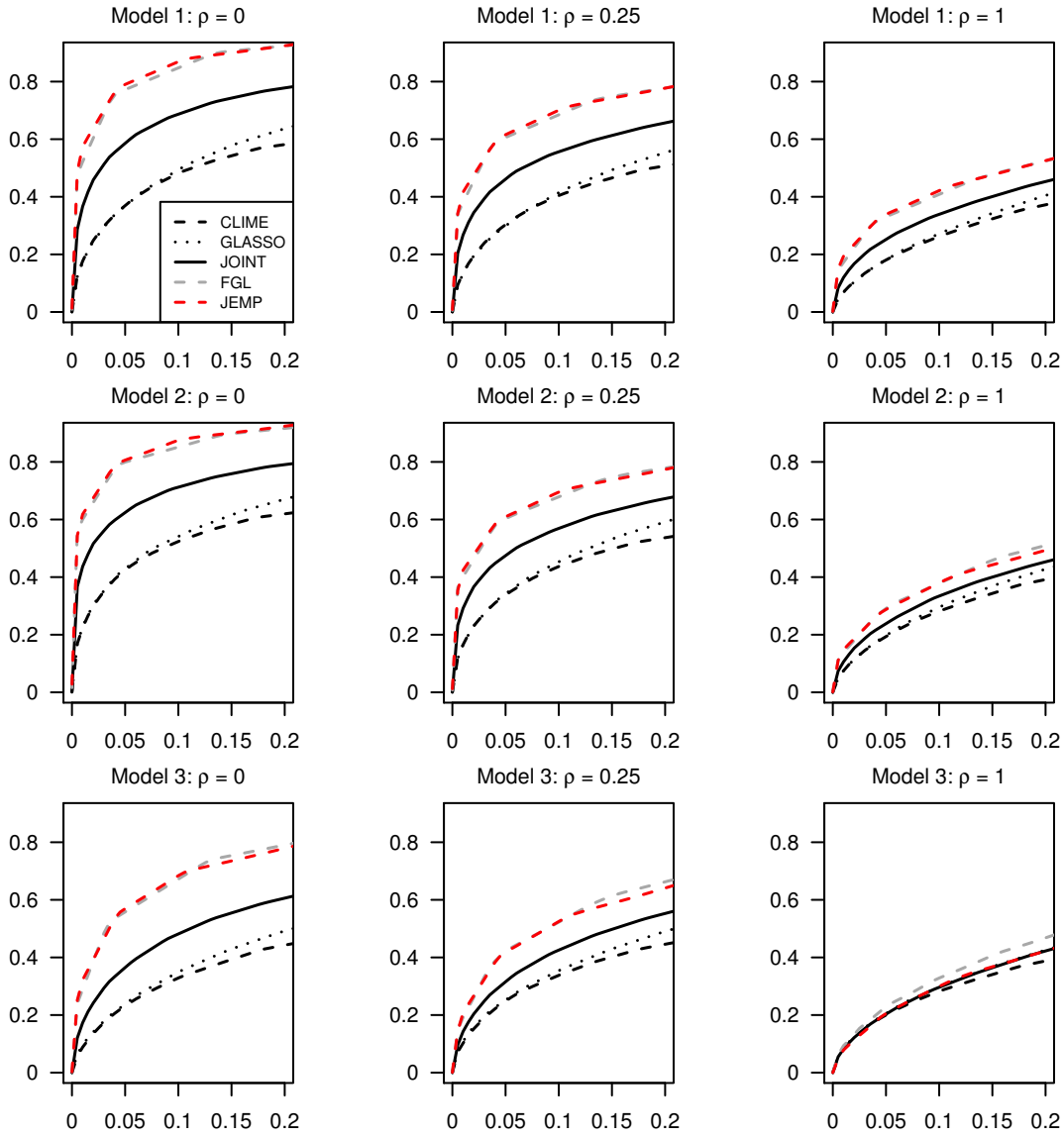


Figure 1: Receiver operating characteristic curves averaged over 50 replications from Gaussian distributions. In each panel, the horizontal and vertical axes are false positive rate and sensitivity respectively. Here, ρ is the ratio of the number of unique nonzero entries to the number of common nonzero entries.

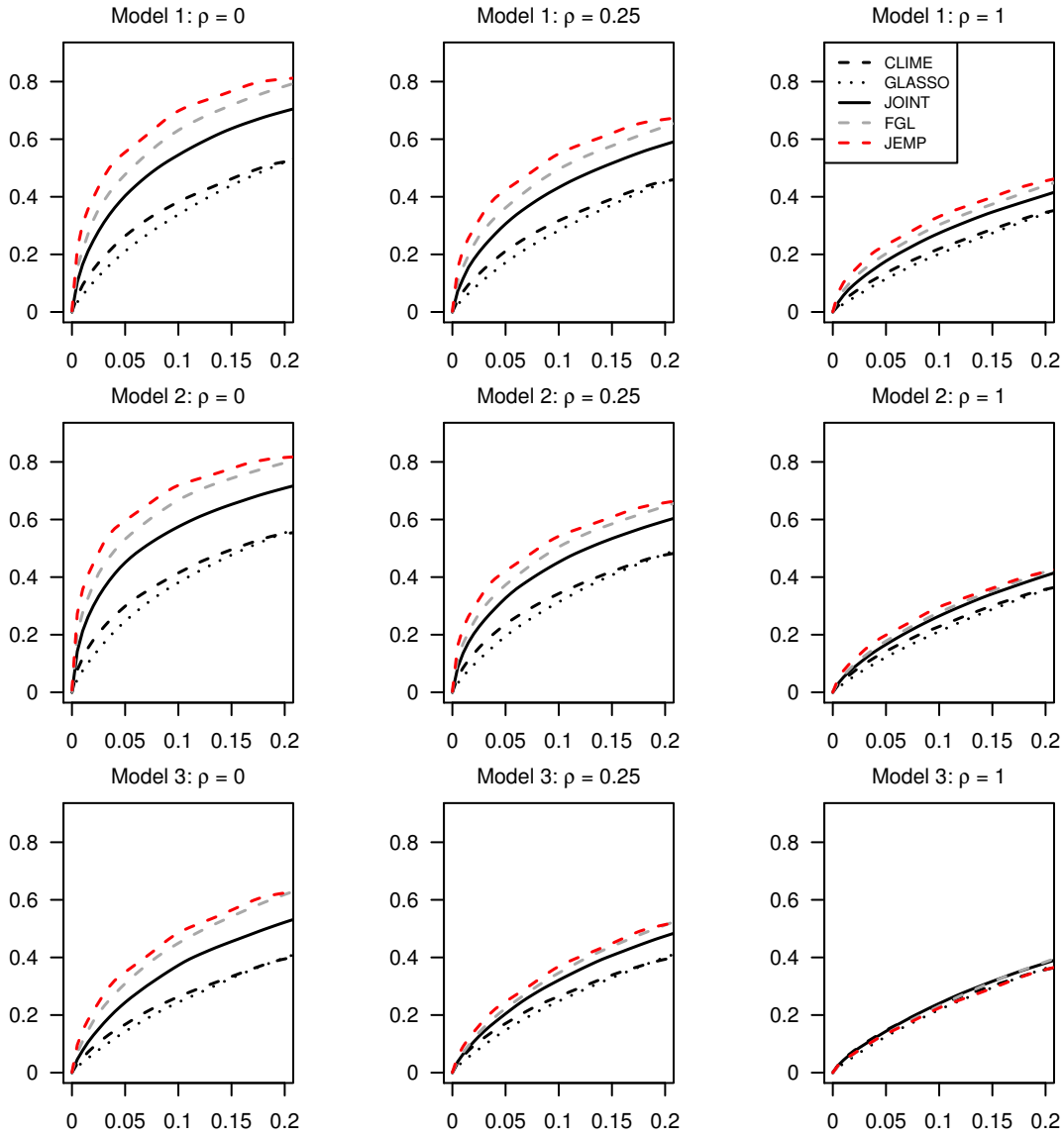


Figure 2: Receiver operating characteristic curves averaged over 50 replications from t_5 distributions. In each panel, the horizontal and vertical axes are false positive rate and sensitivity respectively. Here, ρ is the ratio of the number of unique nonzero entries to the number of common nonzero entries.

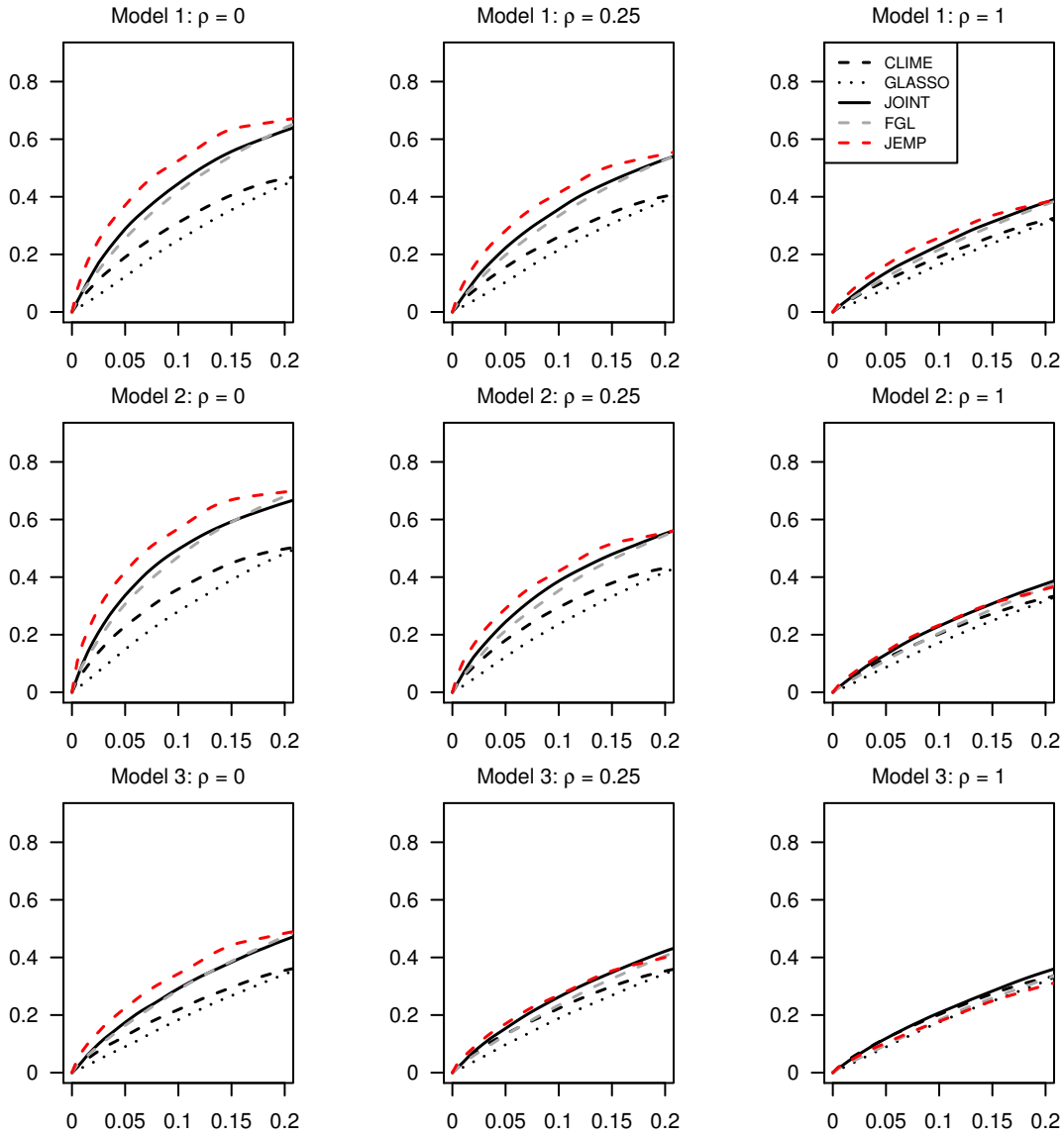


Figure 3: Receiver operating characteristic curves averaged over 50 replications from t_3 distributions. In each panel, the horizontal and vertical axes are false positive rate and sensitivity respectively. Here, ρ is the ratio of the number of unique nonzero entries to the number of common nonzero entries.

Figures 1-3 show the estimated receiver operating characteristic (ROC) curves averaged over 50 replications. In the Gaussian case of Figure 1, JEMP and FGL show similar performance and outperform the others except the case of $\rho = 1$ in Model 3. In Figures 2 and 3 of multivariate t-distributions, it can be observed that JEMP has better ROC curves when $\rho = 0$ for all three models. It also shows better performance than the others when $\rho = 0.25$ for Models 1-2. When $\rho = 1$, all ROC curves move closer together. This is because the true precision matrices become much denser in terms of the number of edges and thus all methods have some difficulty in edge selection. Overall, our proposed JEMP estimator delivers competitive performance in terms of both estimation accuracy and selection.

Note that JEMP and FGL encourage the estimated precision matrices to be similar across all classes. This can be advantageous especially when the true precision matrices have many common values. Therefore, JEMP and FGL can have better performance than JOINT for such problems.

In terms of computational complexity, JEMP can be more intensive than separate estimation methods and JOINT as it involves a pair of tuning parameters (λ_1, λ_2) satisfying $\lambda_1 \leq \lambda_2$. The computational cost of JEMP can be potentially reduced using the ADMM algorithm discussed in Section 4 with a further improved algorithm for the least squares step.

6. Application on Glioblastoma Cancer Data

In this section, we apply our joint method to a Glioblastoma cancer data set. The data set consists of 17814 gene expression levels of 482 GBM patients. The patients were classified into four subtypes, namely, classical, mesenchymal, neural, and proneural with sample sizes of 127, 145, 85, and 125 respectively (Verhaak et al., 2010). These subtypes are shown to be different biologically, while at the same time, share similarities as well since they all belong to GBM cancer. In this application, we consider the signature genes reported by Verhaak et al. (2010). They established 210 signature genes for each subtype, which results 840 signature genes in total. These signature genes are highly distinctive for four subtypes and reported to have good predictive power for subtype prediction. In our analysis, the goal is to produce graphical presentation of relationships among these signature genes in each subtype based on the estimation of the precision matrices. Among the 840 signature genes, we excluded the genes with no subtype information or the genes with missing values. As a result, total 680 genes were included in our analysis. To produce interpretable graphical models using our JEMP estimator, we set the values of the tuning parameters as $\lambda_1 = 0.30$ and $\lambda_2 = 0.40$. JEMP estimated 214 edges shared among all subtypes, 9 edges present only in two subtypes, and 1 edge present only in three subtypes.

The resulting gene networks are shown in Figure 4. The black lines are the edges shared by all subtypes and the thick grey lines are the unique edges present only in two or three subtypes. It is noticeable that most of edges are black lines, which means that they appear in all subtypes. This indicates that the networks of the signature genes reported by Verhaak et al. (2010) may be very similar across all subtypes as they all belong to GBM cancer.

All of the small red network's genes in the upper region belong to the ZNF gene family. This network includes ZNF211, ZNF227, ZNF228, ZNF235, ZNF419, and ZNF671. These are known to be involved in making zinc finger proteins, which are regulatory proteins

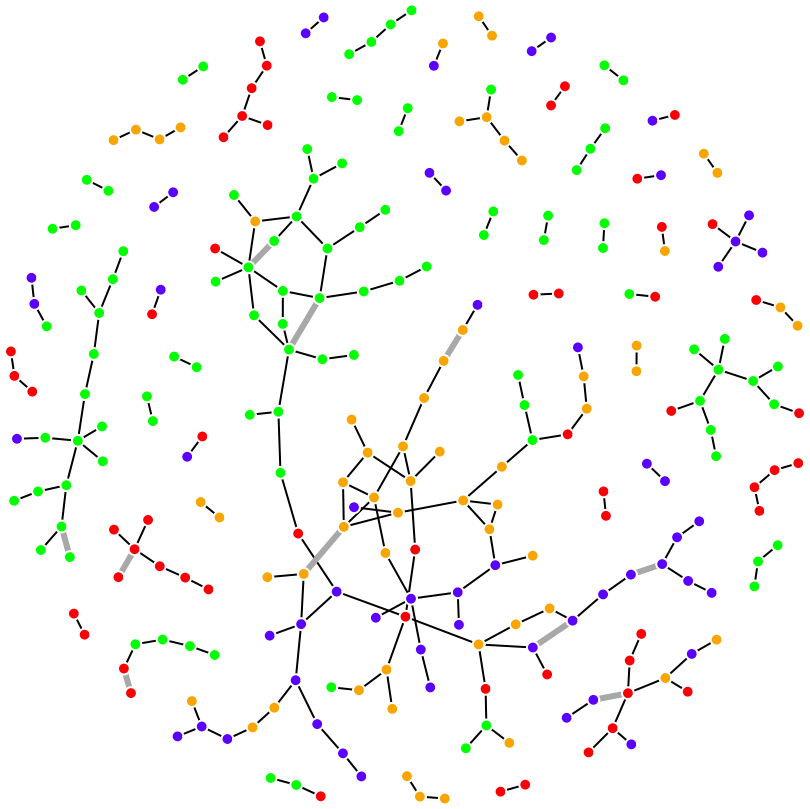


Figure 4: Graphical presentation of conditional dependence structures among genes using our estimator of precision matrices. The black lines are the edges shared in all subtypes and the thick grey lines are the unique edges present only in two or three subtypes. The red, green, blue and orange genes are classical, mesenchymal, proneural and neural genes respectively (Verhaak et al., 2010).

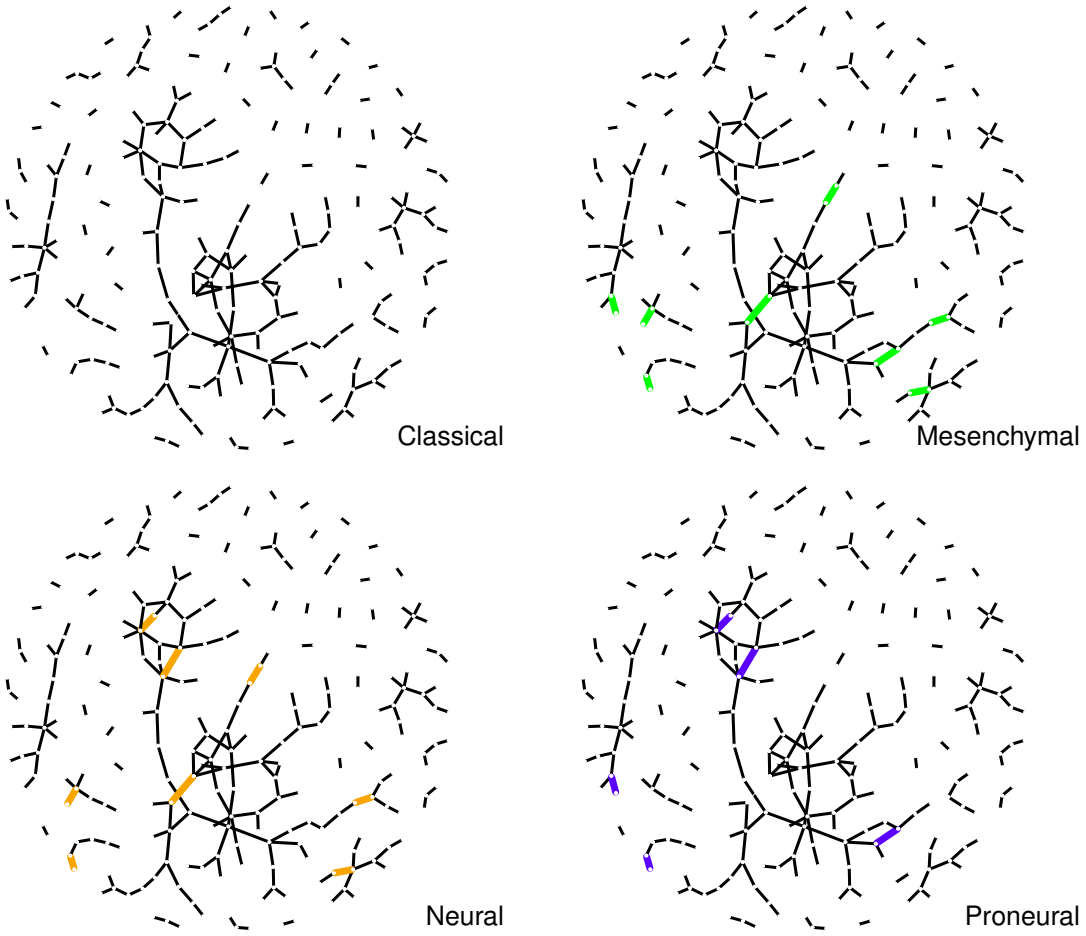


Figure 5: Four gene networks corresponding to four subtypes of the GMB cancer. In each network, the black lines are the edges shared in all subtypes. The colored lines are the edge shared only in two or three subtypes.

that are related to many cellular functions. As they are all involved in the same biological process, it may seem reasonable that this network is shared in all GBM subtypes.

The red genes are signature genes for the classical subtype. Likewise, green, blue and orange genes are the mesenchymal, proneural and neural signature genes respectively. Each class of signature genes tends to have more links with the genes in the same class. This is expected because each class of signature genes is more likely to be highly co-expressed.

Each estimated network for each subtype is depicted in Figure 5. The black lines are the edges shared by all subtypes and the colored lines are the edges appearing only in two or three subtypes. One interesting edge is the one between EGFR and MEOX2. It does not appear in the classical subtype while it is shared by all the other subtypes. EGFR is known to be involved in cell proliferation and Verhaak et al. (2010) demonstrated the essential role of this gene in GBM tumor genesis. Furthermore, high rates of EGFR alteration were claimed in the classical subtype. Therefore, studying the relationship between EGFR and MEOX2 can be an interesting direction for future investigation as only the classical subtype lacks this edge.

There are 9 edges appearing only in two subtypes. These include SCG3 and ACSBG1, GRIK5 and BTBD2, NCF4 and CSTA, IFI30 and BATF, HK3 and SLC11A1, ACSBG1 and SCG3, GPM6A and OLIG2, C1orf61 and CKB, and PPFIA2 and GRM1. It would be also interesting to investigate these relationships further as they are unique only in two subtypes. For example, the edge between OLIG2 and GPM6A does not appear in the proneural subtype while it is shared by Neural and Mesenchymal subtypes. High expression of OLIG2 was observed in the proneural subtype (Verhaak et al., 2010), which can down-regulate the tumor suppressor p21. Therefore, it may be helpful to investigate the relationship between OLIG2 and GPM6A for understanding the effect of OLIG2 in the proneural subtype.

Acknowledgments

The authors would like to thank the Action Editor Professor Francis Bach and three reviewers for their constructive comments and suggestions. The authors were supported in part by NIH/NCI grant R01 CA-149569, NIH/NCI P01 CA-142538, and NSF grant DMS-1407241.

Appendix A.

Write $\Sigma_0^{(g)} = (\sigma_{ij,0}^{(g)})$ and $\hat{\Sigma}^{(g)} = (\hat{\sigma}_{ij}^{(g)})$. Let $m_{j,0}$ and $r_{j,0}^{(g)}$ be the jth columns of M_0 and $R_0^{(g)}$ respectively. Define the jth columns of $\hat{M}$ and $\hat{R}^{(g)}$ as $\hat{m}_j$ and $\hat{r}_j^{(g)}$ respectively. We first state some results established by Cai et al. (2011) in the proof of their Theorem 1.

Lemma 4 *Suppose Condition 1 holds. For any fixed $g = 1, \ldots, G$, with probability greater than $1 - 4p^{-\tau}$,*

$$\max_{ij} |\hat{\sigma}_{ij}^{(g)} - \sigma_{ij,0}^{(g)}| \leq C_0 \left(\frac{\log p}{n} \right)^{1/2},$$

where C_0 is given in Theorem 1.

Proof [Proof of Theorem 1] It follows from Lemma 4 that

$$\max_{ij} |\hat{\sigma}_{ij}^{(g)} - \sigma_{ij,0}^{(g)}| \leq \lambda_2/(3C_M) \quad \text{for all } g = 1, \ldots, G, \tag{6}$$

with probability greater than $1 - 4Gp^{-\tau}$. All following arguments assume (6) holds. First, we have that

$$\begin{aligned}
|(\hat{\Omega}_1^{(g)} - \Omega_0^{(g)})e_j|_\infty &= |\Omega_0^{(g)}(\Sigma_0^{(g)}\hat{\Omega}_1^{(g)} - I)e_j|_\infty \leq ||\Omega_0^{(g)}||_{L_1}|(\Sigma_0^{(g)}\hat{\Omega}_1^{(g)} - I)e_j|_\infty \\
&\leq C_M \left\{ |(\Sigma_0^{(g)} - \hat{\Sigma}^{(g)})\hat{\Omega}_1^{(g)}e_j|_\infty + |(\hat{\Sigma}^{(g)}\hat{\Omega}_1^{(g)} - I)e_j|_\infty \right\} \\
&\leq C_M |\hat{\Omega}_1^{(g)}e_j|_1 |\Sigma_0^{(g)} - \hat{\Sigma}^{(g)}|_\infty + C_M\lambda_2 \\
&\leq |\hat{\Omega}_1^{(g)}e_j|_1 \lambda_2/3 + C_M\lambda_2,
\end{aligned}$$

for all $g = 1, \ldots, G$. Second, note that $\{M_0, R_0^{(1)}, \ldots, R_0^{(G)}\}$ is a feasible solution of (3) as $|I - \hat{\Sigma}^{(g)}(M_0 + R_0^{(g)})|_\infty = |(\Sigma_0^{(g)} - \hat{\Sigma}^{(g)})\Omega_0^{(g)}|_\infty \leq ||\Omega_0^{(g)}||_{L_1}|\Sigma_0^{(g)} - \hat{\Sigma}^{(g)}|_\infty \leq C_M\lambda_2/(3C_M) < \lambda_2$ and $\lambda_1 = \lambda_2$. Therefore, we have that

$$\begin{aligned}
\sum_{g=1}^{G} |(\hat{\Omega}_1^{(g)} - \Omega_0^{(g)})e_j|_\infty &\leq \sum_{g=1}^{G} |\hat{\Omega}_1^{(g)}e_j|_1 \lambda_2/3 + GC_M\lambda_2 \leq G \left\{ |\hat{m}_j|_1 + G^{-1} \sum_{g=1}^{G} |\hat{r}_j^{(g)}|_1 \right\} \lambda_2/3 + GC_M\lambda_2 \\
&\leq G \left\{ |m_{j,0}|_1 + G^{-1} \sum_{g=1}^{G} |r_{j,0}^{(g)}|_1 \right\} \lambda_2/3 + GC_M\lambda_2 \\
&\leq G3C_M\lambda_2/3 + GC_M\lambda_2 = 2GC_M\lambda_2 = 6GC_M^2C_0(\log p/n)^{1/2}.
\end{aligned}$$

By the inequality

$$\max_{ij} \left(\frac{1}{G} \sum_{g=1}^{G} |\hat{\omega}_{ij}^{(g)} - \omega_{ij,0}^{(g)}| \right) \leq \max_j \frac{1}{G} \sum_{g=1}^{G} |(\hat{\Omega}_1^{(g)} - \Omega_0^{(g)})e_j|_\infty \leq 6C_M^2C_0 \left(\frac{\log p}{n} \right)^{1/2},$$

the proof is completed. ■

Lemma 5 *With probability greater than* $1 - 2(1 + G)p^{-\tau}$*, the following holds:*

$$\max_{ij} |\sum_{g=1}^{G} (\hat{\sigma}_{ij}^{(g)} - \sigma_{ij,0}^{(g)})| \leq C_0 \left(\frac{G \log p}{n} \right)^{1/2}.$$

Proof We adopt a similar technique used in Cai et al. (2011) for the proof of their Theorem 1. Without loss of generality, we assume that $\mu_i^{(g)} = 0$ for all i and g. Let $y_{kij}^{(g)} := x_{ki}^{(g)}x_{kj}^{(g)} - E(x_{ki}^{(g)}x_{kj}^{(g)})$. Define $\bar{x}_i^{(g)} := \sum_{k=1}^{n} x_{ki}^{(g)}/n; i = 1, \ldots, p, g = 1, \ldots, G$. Then $\sum_{g=1}^{G}(\hat{\sigma}_{ij}^{(g)} - \sigma_{ij,0}^{(g)}) = \sum_{g=1}^{G} \left(\sum_{k=1}^{n} y_{kij}^{(g)}/n - \bar{x}_i^{(g)}\bar{x}_j^{(g)} \right)$. Let $t := \eta(\log p)^{1/2}(nG)^{-1/2}$ and $C_1 := 2 + \tau + \eta^{-1}K^2$. Using the Markov's inequality and the inequality $|\exp(s) - 1 - s| \leq s^2 \exp\{\max(s, 0)\}$ for

any $s \in \mathcal{R}$, we can show that

$$\begin{aligned}
&\text{pr}\left\{\frac{1}{n}\sum_{g=1}^{G}\sum_{k=1}^{n} y_{kij}^{(\text{g})} \geq \eta^{-1}C_1\left(\frac{G\log p}{n}\right)^{1/2}\right\} \\
&= \text{pr}\left\{\sum_{g=1}^{G}\sum_{k=1}^{n} y_{kij}^{(\text{g})} \geq \eta^{-1}C_1\left(nG\log p\right)^{1/2}\right\} \\
&\leq \exp\left\{-t\eta^{-1}C_1(nG\log p)^{1/2}\right\} E\left\{\exp\left(t\sum_{g=1}^{G}\sum_{k=1}^{n} y_{kij}^{(\text{g})}\right)\right\} \\
&= \exp\left\{-C_1\log p\right\}\prod_{g=1}^{G}\prod_{k=1}^{n} E\left\{\exp(ty_{kij}^{(\text{g})})\right\} \\
&= \exp\left[-C_1\log p + \sum_{g=1}^{G} n\log\left\{E\left(e^{ty_{kij}^{(\text{g})}}\right)\right\}\right] \\
&\leq \exp\left[-C_1\log p + \sum_{g=1}^{G} n\left\{E\left(e^{ty_{kij}^{(\text{g})}}\right)-1\right\}\right] \\
&= \exp\left[-C_1\log p + \sum_{g=1}^{G} n\left\{E\left(e^{ty_{kij}^{(\text{g})}}-ty_{kij}^{(\text{g})}-1\right)\right\}\right] \\
&\leq \exp\left\{-C_1\log p + \sum_{g=1}^{G} nt^2 E\left({y_{kij}^{(\text{g})}}^2 e^{|ty_{kij}^{(\text{g})}|}\right)\right\} \\
&\leq \exp\left\{-C_1\log p + \sum_{g=1}^{G} (\eta G)^{-1}K^2\log p\right\}. \qquad (7)
\end{aligned}$$

The last inequality (7) holds since

$$nt^2 E\left({y_{kij}^{(\text{g})}}^2 e^{|ty_{kij}^{(\text{g})}|}\right) = (\eta G)^{-1}(\log p)E\left\{\left(\eta^{3/2}|y_{kij}^{(\text{g})}|\right)^2 e^{t|y_{kij}^{(\text{g})}|}\right\}$$

and

$$\begin{aligned}
E\left\{\left(\eta^{3/2}|y_{kij}^{(\text{g})}|\right)^2 e^{t|y_{kij}^{(\text{g})}|}\right\} &\leq E\left\{e^{\eta^{3/2}|y_{kij}^{(\text{g})}|e^{t|y_{kij}^{(\text{g})}|}}\right\} \leq E\left\{e^{\eta^{3/2}|y_{kij}^{(\text{g})}|e^{\eta^{3/2}|y_{kij}^{(\text{g})}|}}\right\} \\
&\leq E\left\{e^{\eta|y_{kij}^{(\text{g})}|}\right\} \leq E\left\{e^{\eta|x_{ki}^{(\text{g})}x_{kj}^{(\text{g})}|+\eta E\left(|x_{ki}^{(\text{g})}x_{kj}^{(\text{g})}|\right)}\right\} \\
&\leq \left\{E\left(e^{\eta|x_{ki}^{(\text{g})}x_{kj}^{(\text{g})}|}\right)\right\}^2 \leq \left\{E\left(e^{\eta{x_{ki}^{(\text{g})}}^2/2+\eta{x_{kj}^{(\text{g})}}^2/2}\right)\right\}^2 \\
&\leq E\left(e^{\eta{x_{ki}^{(\text{g})}}^2}\right)E\left(e^{\eta{x_{kj}^{(\text{g})}}^2}\right) \leq K^2.
\end{aligned}$$

From (7), it follows that

$$\text{pr}\left\{\frac{1}{n}\sum_{g=1}^{G}\sum_{k=1}^{n} y_{kij}^{(\text{g})} \geq \eta^{-1}C_1\left(\frac{G\log p}{n}\right)^{1/2}\right\} \leq \exp\left\{-C_1\log p + \eta^{-1}K^2\log p\right\} \leq p^{-(\tau+2)}.$$

Therefore, we have

$$\mathrm{pr}\left\{\max_{ij}\left|\frac{1}{n}\sum_{g=1}^{G}\sum_{k=1}^{n}y_{kij}^{(\mathrm{g})}\right|\geq\eta^{-1}C_1\left(\frac{G\log p}{n}\right)^{1/2}\right\}\leq 2p^{-\tau}. \tag{8}$$

Next, let $C_2=2+\tau+\eta^{-1}(eK)^2$. Cai et al. (2011) showed in the proof of their Theorem 1 that

$$\mathrm{pr}\left(\max_{ij}|\bar{x}_i^{(\mathrm{g})}\bar{x}_j^{(\mathrm{g})}|\geq\eta^{-2}C_2^2\log p/n\right)\leq 2p^{-\tau-1}.$$

Using this result, we have that

$$\begin{aligned}\mathrm{pr}\left(\max_{ij}|\sum_{g=1}^{G}\bar{x}_i^{(\mathrm{g})}\bar{x}_j^{(\mathrm{g})}|\geq\eta^{-2}C_2^2G\log p/n\right)&\leq\mathrm{pr}\left(\sum_{g=1}^{G}\max_{ij}|\bar{x}_i^{(\mathrm{g})}\bar{x}_j^{(\mathrm{g})}|\geq\eta^{-2}C_2^2G\log p/n\right)\\&\leq\sum_{g=1}^{G}\mathrm{pr}\left(\max_{ij}|\bar{x}_i^{(\mathrm{g})}\bar{x}_j^{(\mathrm{g})}|\geq\eta^{-2}C_2^2\log p/n\right)\\&\leq\sum_{g=1}^{G}2p^{-\tau-1}\leq 2Gp^{-\tau}\end{aligned} \tag{9}$$

By (8), (9) and the inequality $C_0>\eta^{-1}C_1+\eta^{-2}C_2^2(G\log p/n)^{1/2}$, we see that

$$\begin{aligned}&\mathrm{pr}\left\{\max_{ij}|\sum_{g=1}^{G}(\hat{\sigma}_{ij}^{(\mathrm{g})}-\sigma_{ij,0}^{(\mathrm{g})})|\geq C_0\left(\frac{G\log p}{n}\right)^{1/2}\right\}\\&\leq\mathrm{pr}\left\{\max_{ij}\left|\frac{1}{n}\sum_{g=1}^{G}\sum_{k=1}^{n}y_{kij}^{(\mathrm{g})}\right|\geq\eta^{-1}C_1\left(\frac{G\log p}{n}\right)^{1/2}\right\}\\&+\mathrm{pr}\left(\max_{ij}|\sum_{g=1}^{G}\bar{x}_i^{(\mathrm{g})}\bar{x}_j^{(\mathrm{g})}|\geq\eta^{-2}C_2^2G\log p/n\right)\\&\leq 2(1+G)p^{-\tau}.\end{aligned}$$

The proof is completed. ■

Proof [Proof of Theorem 2] By Lemma 4 and 5, we see that

$$\max_{ij}|\sum_{g=1}^{G}(\hat{\sigma}_{ij}^{(\mathrm{g})}-\sigma_{ij,0}^{(\mathrm{g})})|\leq C_0\left(\frac{G\log p}{n}\right)^{1/2}\text{and}\max_{ij}|\hat{\sigma}_{ij}^{(\mathrm{g})}-\sigma_{ij,0}^{(\mathrm{g})}|\leq C_0\left(\frac{\log p}{n}\right)^{1/2}, \tag{10}$$

for all $g=1,\ldots,G$ with probability greater than $1-2(1+3G)p^{-\tau}$. All following arguments assume (10) holds. Note that $\{M_0,R_0^{(1)},\ldots,R_0^{(\mathrm{G})}\}$ is a feasible solution of (3) as

$$\begin{aligned}|I-\hat{\Sigma}^{(\mathrm{g})}(M_0+R_0^{(\mathrm{g})})|_\infty=|(\Sigma_0^{(\mathrm{g})}-\hat{\Sigma}^{(\mathrm{g})})\Omega_0^{(\mathrm{g})}|_\infty&\leq||\Omega_0^{(\mathrm{g})}||_{L_1}|\Sigma_0^{(\mathrm{g})}-\hat{\Sigma}^{(\mathrm{g})}|_\infty\\&\leq C_MC_0(\log p/n)1/2=\lambda_2\end{aligned}$$

and

$$
\begin{aligned}
&|G^{-1}\sum_{g=1}^{G}\left\{I-\hat{\Sigma}^{(g)}(M_0+R_0^{(g)})\right\}|_\infty \\
&\le |G^{-1}\sum_{g=1}^{G}(\Sigma_0^{(g)}-\hat{\Sigma}^{(g)})M_0|_\infty + |G^{-1}\sum_{g=1}^{G}(\Sigma_0^{(g)}-\hat{\Sigma}^{(g)})R_0^{(g)}|_\infty \\
&\le ||M_0||_{L_1}|G^{-1}\sum_{g=1}^{G}(\Sigma_0^{(g)}-\hat{\Sigma}^{(g)})|_\infty + G^{-1}\sum_{g=1}^{G}||R_0^{(g)}||_{L_1}|\Sigma_0^{(g)}-\hat{\Sigma}^{(g)}|_\infty \\
&\le C_M C_0\left\{\log p/(nG)\right\}^{1/2} + C_R C_0\left\{\log p/(nG)\right\}^{1/2} = \lambda_1.
\end{aligned}
$$

Now, we find an upper bound of $|G(\hat{M}-M_0)e_j|_\infty = |\sum_{g=1}^{G}(\hat{\Omega}_1^{(g)}-\Omega_0^{(g)})e_j|_\infty$. In particular, we use

$$
|\sum_{g=1}^{G}(\hat{\Omega}_1^{(g)}-\Omega_0^{(g)})e_j|_\infty \le |\sum_{g=1}^{G}\Omega_0^{(g)}(\Sigma_0^{(g)}-\hat{\Sigma}^{(g)})\hat{\Omega}_1^{(g)}e_j|_\infty + |\sum_{g=1}^{G}\Omega_0^{(g)}(\hat{\Sigma}^{(g)}\hat{\Omega}_1^{(g)}-I)e_j|_\infty. \quad (11)
$$

First, consider the first term in the right-hand side of (11). We can show that

$$
\begin{aligned}
|\sum_{g=1}^{G}\Omega_0^{(g)}(\Sigma_0^{(g)}-\hat{\Sigma}^{(g)})\hat{\Omega}_1^{(g)}e_j|_\infty &\le |\sum_{g=1}^{G}M_0(\Sigma_0^{(g)}-\hat{\Sigma}^{(g)})\hat{m}_j|_\infty + |\sum_{g=1}^{G}M_0^{(g)}(\Sigma_0^{(g)}-\hat{\Sigma}^{(g)})\hat{r}_j^{(g)}|_\infty \\
&\quad + |\sum_{g=1}^{G}R_0^{(g)}(\Sigma_0^{(g)}-\hat{\Sigma}^{(g)})\hat{m}_j|_\infty + |\sum_{g=1}^{G}R_0^{(g)}(\Sigma_0^{(g)}-\hat{\Sigma}^{(g)})\hat{r}_j^{(g)}|_\infty \\
&\le ||M_0||_{L_1}\left\{|\sum_{g=1}^{G}(\Sigma_0^{(g)}-\hat{\Sigma}^{(g)})|_\infty|\hat{m}_j|_1 + \sum_{g=1}^{G}|\Sigma_0^{(g)}-\hat{\Sigma}^{(g)}|_\infty|\hat{r}_j^{(g)}|_1\right\} \\
&\quad + \sum_{g=1}^{G}|R_0^{(g)}(\Sigma_0^{(g)}-\hat{\Sigma}^{(g)})|_\infty|\hat{m}_j|_1 + \sum_{g=1}^{G}|R_0^{(g)}(\Sigma_0^{(g)}-\hat{\Sigma}^{(g)})|_\infty|\hat{r}_j^{(g)}|_1.
\end{aligned}
$$

Using the assumptions $||R_0^{(g)}||_{L_1} \le C_R$ and $\sum_{g=1}^{G}||R_0^{(g)}||_{L_1} \le G^{1/2}C_R$, we have

$$
\begin{aligned}
|\sum_{g=1}^{G}\Omega_0^{(g)}(\Sigma_0^{(g)}-\hat{\Sigma}^{(g)})\hat{\Omega}_1^{(g)}e_j|_\infty &\le C_M C_0(G\log p/n)^{1/2}|\hat{m}_j|_1 + C_M C_0(\log p/n)^{1/2}\sum_{g=1}^{G}|\hat{r}_j^{(g)}|_1 \\
&\quad + C_R C_0(G\log p/n)^{1/2}|\hat{m}_j|_1 + C_R C_0(\log p/n)^{1/2}\sum_{g=1}^{G}|\hat{r}_j^{(g)}|_1 \\
&\le C_0(C_M+C_R)(G\log p/n)^{1/2}(|\hat{m}_j|_1 + G^{-1/2}\sum_{g=1}^{G}|\hat{r}_j^{(g)}|_1) \\
&\le C_0(C_M+C_R)(G\log p/n)^{1/2}(|m_{j,0}|_1 + G^{-1/2}\sum_{g=1}^{G}|r_{j,0}^{(g)}|_1) \\
&\le C_0(C_M+C_R)^2(G\log p/n)^{1/2}. \quad (12)
\end{aligned}
$$

For the second term in the right-hand side of (11), note that

$$
\begin{aligned}
&|\sum_{g=1}^{G} \Omega_0^{(g)}(\hat{\Sigma}^{(g)}\hat{\Omega}_1^{(g)} - I)e_j|_\infty \\
&\leq |\sum_{g=1}^{G} M_0(\hat{\Sigma}^{(g)}\hat{\Omega}^{(g)} - I)e_j|_\infty + |\sum_{g=1}^{G} R_0^{(g)}(\hat{\Sigma}^{(g)}\hat{\Omega}^{(g)} - I)e_j|_\infty \\
&\leq ||M_0||_{L_1}|\sum_{g=1}^{G}(\hat{\Sigma}^{(g)}\hat{\Omega}^{(g)} - I)e_j|_\infty + \sum_{g=1}^{G}||R_0^{(g)}||_{L_1}|(\hat{\Sigma}^{(g)}\hat{\Omega}^{(g)} - I)e_j|_\infty \\
&\leq C_M\lambda_1 + G^{1/2}C_R\lambda_2 = C_0C_M(C_M + 2C_R)(G\log p/n)^{1/2}. \qquad (13)
\end{aligned}
$$

By (11), (12), (13) and the equality $|\hat{M} - M_0|_\infty = \max_j |(\hat{M} - M_0)e_j|_\infty$, we have

$$|\hat{M} - M_0|_\infty \leq C_0(2C_M^2 + 4C_MC_R + C_R^2)\left(\frac{\log p}{nG}\right)^{1/2}.$$

The proof is completed. ■

Proof [Proof of Theorem 3] By Theorem 1, we see that

$$\max_{ij}\sum_{g=1}^{G}|\hat{\omega}_{ij}^{(g)} - \omega_{ij,0}^{(g)}| \leq 2GC_M\lambda_2 \leq \delta_n, \qquad (14)$$

with probability greater than $1 - 4Gp^{-\tau}$. We show that $S_0 = \hat{S}$ when (14) holds. For any $(i, j, g) \notin S_0$, we have $|\hat{\omega}_{ij}^{(g)}| = |\hat{\omega}_{ij}^{(g)} - \omega_{ij,0}^{(g)}| \leq \sum_{g=1}^{G}|\hat{\omega}_{ij}^{(g)} - \omega_{ij,0}^{(g)}| \leq \delta_n$. Therefore, we see $\tilde{\omega}_{ij}^{(g)} = 0$, which implies $\hat{S} \subset S_0$. On the other hand, for any $(i, j, g) \in S_0$, we have $|\hat{\omega}_{ij}^{(g)}| \geq |\omega_{ij,0}^{(g)}| - |\hat{\omega}_{ij}^{(g)} - \omega_{ij,0}^{(g)}| \geq |\omega_{ij,0}^{(g)}| - \sum_{g=1}^{G}|\hat{\omega}_{ij}^{(g)} - \omega_{ij,0}^{(g)}| > \delta_n$. Therefore, we see that $\tilde{\omega}_{ij}^{(g)} \neq 0$, which implies $S_0 \subset \hat{S}$. In summary, we see that $S_0 = \hat{S}$ if (14) holds, which implies that $\text{pr}(S_0 = \hat{S}) \geq \text{pr}(\max_{ij}\sum_{g=1}^{G}|\hat{\omega}_{ij}^{(g)} - \omega_{ij,0}^{(g)}| \leq \delta_n)$. ■

References

Onureena Banerjee, Laurent El Ghaoui, and Alexandre d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *Journal of Machine Learning Research*, 9:485–516, 2008.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, 2004.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3:1–122, 2010.

Tony Cai, Weidong Liu, and Xi Luo. A constrained l_1 minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association*, 106:594–607, 2011.

Patrick Danaher, Pei Wang, and Daniela M. Witten. The joint graphical lasso for inverse covariance estimation across multiple classes. *Journal of the Royal Statistical Society, Series B*, 76:373–379, 2014.

Theodoros Evgeniou and Massimiliano Pontil. Regularized multitask learning. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 109–117, Seattle, Washington, 2004.

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

Jianqing Fan, Yang Feng, and Yichao Wu. Network exploration via the adaptive lasso and scad penalties. *The Annals of Applied Statistics*, 3:521–541, 2009.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9:432–441, 2008.

Jian Guo, Elizaveta Levina, George Michailidis, and Ji Zhu. Joint estimation of multiple graphical models. *Biometrika*, 98:1–15, 2011.

Jean Honorio and Dimitris Samaras. Simultaneous and group-sparse multi-task learning of gaussian graphical models. *arXiv:1207.4255*, 2012.

Clifford Lam and Jianqing Fan. Sparsistency and rates of convergence in large covariance matrix estimation. *The Annals of Statistics*, 37:4254–4278, 2009.

Wonyul Lee, Ying Du, Wei Sun, David Neil Hayes, and Yufeng Liu. Multiple response regression for gaussian mixture models with known labels. *Statistical Analysis and Data Mining*, 5:493–508, 2012.

Nicolai Meinshausen and Peter Bühlmann. High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, 34:1436–1462, 2006.

Haotian Pang, Han Liu, and Robert Vanderbei. *fastclime: A fast solver for parameterized lp problems and constrained l_1-minimization approach to sparse precision matrix estimation*, 2014. R package version 1.2.4.

Jie Peng, Pei Wang, Nengfeng Zhou, and Ji Zhu. Partial correlation estimation by joint sparse regression models. *Journal of the American Statistical Association*, 104:735–746, 2009.

Pradeep Ravikumar, Martin J. Wainwright, Garvesh Raskutti, and Bin Yu. High-dimensional covariance estimation by minimizing l_1-penalized log-determinant divergence. *Electronic Journal of Statistics*, 5:935–980, 2011.

Adam J. Rothman, Peter J. Bickel, Elizaveta Levina, and Ji Zhu. Sparse permutation invariant covariance estimation. *Electronic Journal of Statistics*, 2:494–515, 2008.

The Cancer Genome Atlas Research Network. Comprehensive genomic characterization defines human glioblastoma genes and core pathways. *Nature*, 455:1061–1068, 2008.

Roel G.W. Verhaak, Katherine A. Hoadley, Elizabeth Purdom, Victoria Wang, Yuan Qi, Matthew D. Wilkerson, C. Ryan Miller, Li Ding, Todd Golub, Jill P. Mesirov, Gabriele Alexe, Michael Lawrence, Michael OKelly, Pablo Tamayo, Barbara A. Weir, Stacey Gabriel, Wendy Winckler, Supriya Gupta, Lakshmi Jakkula, Heidi S. Feiler, J. Graeme Hodgson, C. David James, Jann N. Sarkaria, Cameron Brennan, Ari Kahn, Paul T. Spellman, Richard K. Wilson, Terence P. Speed, Joe W. Gray, Matthew Meyerson, Gad Getz, Charles M. Perou, D. Neil Hayes, and The Cancer Genome Atlas Research Network. Integrated genomic analysis identifies clinically relevant subtypes of glioblastoma characterized by abnormalities in PDGFRA, IDH1, EGFR, and NF1. *Cancer Cell*, 17: 98–110, 2010.

Ming Yuan. High dimensional inverse covariance matrix estimation via linear programming. *Journal of Machine Learning Research*, 11:2261–2286, 2010.

Ming Yuan and Yi Lin. Model selection and estimation in the gaussian graphical model. *Biometrika*, 94:19–35, 2007.

Journal of Machine Learning Research 16 (2015) 1063-1101 Submitted 1/14; Revised 9/14; Published 5/15

Lasso Screening Rules via Dual Polytope Projection

Jie Wang JWANGUMI@UMICH.EDU
Department of Computational Medicine and Bioinformatics
University of Michigan
Ann Arbor, MI 48109-2218, USA
Peter Wonka PETER.WONKA@ASU.EDU
Department of Computer Science and Engineering
Arizona State University
Tempe, AZ 85287-8809, USA
Jieping Ye JPYE@UMICH.EDU
Department of Computational Medicine and Bioinformatics
Department of Electrical Engineering and Computer Science
University of Michigan
Ann Arbor, MI 48109-2218, USA

Editor: Hui Zou

Abstract

Lasso is a widely used regression technique to find sparse representations. When the dimension of the feature space and the number of samples are extremely large, solving the Lasso problem remains challenging. To improve the efficiency of solving large-scale Lasso problems, El Ghaoui and his colleagues have proposed the SAFE rules which are able to quickly identify the inactive predictors, i.e., predictors that have 0 components in the solution vector. Then, the inactive predictors or features can be removed from the optimization problem to reduce its scale. By transforming the standard Lasso to its dual form, it can be shown that the inactive predictors include the set of inactive constraints on the optimal dual solution. In this paper, we propose an efficient and effective screening rule via Dual Polytope Projections (DPP), which is mainly based on the uniqueness and nonexpansiveness of the optimal dual solution due to the fact that the feasible set in the dual space is a convex and closed polytope. Moreover, we show that our screening rule can be extended to identify inactive groups in group Lasso. To the best of our knowledge, there is currently no exact screening rule for group Lasso. We have evaluated our screening rule using synthetic and real data sets. Results show that our rule is more effective in identifying inactive predictors than existing state-of-the-art screening rules for Lasso.

Keywords: lasso, safe screening, sparse regularization, polytope projection, dual formulation, large-scale optimization

1. Introduction

Data with various structures and scales comes from almost every aspect of daily life. To effectively extract patterns in the data and build interpretable models with high prediction accuracy is always desirable. One popular technique to identify important explanatory features is by sparse regularization. For instance, consider the widely used ℓ_1-regularized

least squares regression problem known as Lasso (Tibshirani, 1996). The most appealing property of Lasso is the sparsity of the solutions, which is equivalent to feature selection. Suppose we have N observations and p features. Let $\mathbf{y}$ denote the N dimensional response vector and $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p]$ be the $N \times p$ feature matrix. Let $\lambda \geq 0$ be the regularization parameter. The Lasso problem is formulated as the following optimization problem:

$$\inf_{\beta \in \mathbb{R}^p} \frac{1}{2} \|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \lambda \|\beta\|_1. \tag{1}$$

Lasso has achieved great success in a wide range of applications (Chen et al., 2001; Candès, 2006; Zhao and Yu, 2006; Bruckstein et al., 2009; Wright et al., 2010) and in recent years many algorithms have been developed to efficiently solve the Lasso problem (Efron et al., 2004; Kim et al., 2007; Park and Hastie, 2007; Donoho and Tsaig, 2008; Friedman et al., 2007; Becker et al., 2010; Friedman et al., 2010). However, when the dimension of feature space and the number of samples are very large, solving the Lasso problem remains challenging because we may not even be able to load the data matrix into main memory. The idea of *screening* has been shown very promising in solving Lasso for large-scale problems. Essentially, screening aims to quickly identify the *inactive features* that have 0 components in the solution and then remove them from the optimization. Therefore, we can work on a reduced feature matrix to solve the Lasso problem, which may lead to substantial savings in computational cost and memory usage.

Existing screening methods for Lasso can be roughly divided into two categories: the *Heuristic Screening Methods* and the *Safe Screening Methods*. As the name indicated, the heuristic screening methods can not guarantee that the discarded features have zero coefficients in the solution vector. In other words, they may mistakenly discard the active features which have nonzero coefficients in the sparse representations. Well-known heuristic screening methods for Lasso include SIS (Fan and Lv, 2008) and strong rules (Tibshirani et al., 2012). SIS is based on the associations between features and the prediction task, but not from an optimization point of view. Strong rules rely on the assumption that the absolute values of the inner products between features and the residue are *nonexpansive* (Bauschke and Combettes, 2011) with respect to the parameter values. Notice that, in real applications, this assumption is not always true. In order to ensure the correctness of the solutions, strong rules check the KKT conditions for violations. In case of violations, they weaken the screened set and repeat this process. In contrast to the heuristic screening methods, the safe screening methods for Lasso can guarantee that the discarded features are absent from the resulting sparse models. Existing safe screening methods for Lasso includes SAFE (El Ghaoui et al., 2012) and DOME (Xiang et al., 2011), which are based on an estimation of the dual optimal solution. The key challenge of searching for effective safe screening rules is how to accurately estimate the dual optimal solution. The more accurate the estimation is, the more effective the resulting screening rule is in discarding the inactive features. Moreover, Xiang et al. (2011) have shown that the SAFE rule for Lasso can be read as a special case of their testing rules.

In this paper, we develop novel efficient and effective screening rules for the Lasso problem; our screening rules are safe in the sense that no active features will be discarded. As the name indicated (DPP), the proposed approaches heavily rely on the geometric properties of the Lasso problem. Indeed, the dual problem of problem (1) can be formulated as

a projection problem. More specifically, the **d**ual optimal solution of the Lasso problem is the **p**rojection of the scaled response vector onto a nonempty closed and convex **p**olytope (the feasible set of the dual problem). This nice property provides us many elegant approaches to accurately estimate the dual optimal solutions, e.g., nonexpansiveness, firmly nonexpansiveness (Bauschke and Combettes, 2011). In fact, the estimation of the dual optimal solution in DPP is a direct application of the nonexpansiveness of the projection operators. Moreover, by further exploiting the properties of the projection operators, we can significantly improve the estimation of the dual optimal solution. Based on this estimation, we develop the so called *enhanced DPP* (EDPP) rules which are able to detect far more inactive features than DPP. Therefore, the speedup gained by EDPP is much higher than the one by DPP.

In real applications, the optimal parameter value of λ is generally unknown and needs to be estimated. To determine an appropriate value of λ, commonly used approaches such as cross validation and stability selection involve solving the Lasso problems over a grid of tuning parameters $\lambda_1 > \lambda_2 > \ldots > \lambda_\mathcal{K}$. Thus, the process can be very time consuming. To address this challenge, we develop the sequential version of the DPP families. Briefly speaking, for the Lasso problem, suppose we are given the solution $\beta^*(\lambda_{k-1})$ at λ_{k-1}. We then apply the screening rules to identify the inactive features of problem (1) at λ_k by making use of $\beta^*(\lambda_{k-1})$. The idea of the sequential screening rules is proposed by El Ghaoui et al. (2012) and Tibshirani et al. (2012) and has been shown to be very effective for the aforementioned scenario. In Tibshirani et al. (2012), the authors demonstrate that the sequential strong rules are very effective in discarding inactive features especially for very small parameter values and achieve the state-of-the-art performance. However, in contrast to the recursive SAFE (the sequential version of SAFE rules) and the sequential version of DPP rules, it is worthwhile to mention that the sequential strong rules may mistakenly discard active features because they are heuristic methods. Moreover, it is worthwhile to mention that, for the existing screening rules including SAFE and strong rules, the basic versions are usually special cases of their sequential versions, and the same applies to our DPP and EDPP rules. For the DOME rule (Xiang et al., 2011), it is unclear whether its sequential version exists.

The rest of this paper is organized as follows. We present the family of DPP screening rules, i.e., DPP and EDPP, in detail for the Lasso problem in Section 2. Section 3 extends the idea of DPP screening rules to identify inactive groups in group Lasso (Yuan and Lin, 2006). We evaluate our screening rules on synthetic and real data sets from many different applications. In Section 4, the experimental results demonstrate that our rules are more effective in discarding inactive features than existing state-of-the-art screening rules. We show that the efficiency of the solver can be improved by *several orders of magnitude* with the enhanced DPP rules, especially for the high-dimensional data sets (notice that, the screening methods can be integrated with any existing solvers for the Lasso problem). Some concluding remarks are given in Section 5.

2. Screening Rules for Lasso via Dual Polytope Projections

In this section, we present the details of the proposed DPP and EDPP screening rules for the Lasso problem. We first review some basics of the dual problem of Lasso including its

geometric properties in Section 2.1; we also briefly discuss some basic guidelines for developing safe screening rules for Lasso. Based on the geometric properties discussed in Section 2.1, we then develop the basic DPP screening rule in Section 2.2. As a straightforward extension in dealing with the model selection problems, we also develop the sequential version of DPP rules. In Section 2.3, by exploiting more geometric properties of the dual problem of Lasso, we further improve the DPP rules by developing the so called *enhanced DPP* (EDPP) rules. The EDPP screening rules significantly outperform DPP rules in identifying the inactive features for the Lasso problem.

2.1 Basics

Different from Xiang et al. (2011), we do not assume $\mathbf{y}$ and all $\mathbf{x}_i$ have unit length.The dual problem of problem (1) takes the form of (to make the paper self-contained, we provide the detailed derivation of the dual form in the appendix):

$$\sup_{\theta} \quad \left\{ \frac{1}{2}\|\mathbf{y}\|_2^2 - \frac{\lambda^2}{2}\left\|\theta - \frac{\mathbf{y}}{\lambda}\right\|_2^2 : \ |\mathbf{x}_i^T\theta| \leq 1, \ i = 1, 2, \ldots, p \right\}, \tag{2}$$

where θ is the dual variable. For notational convenience, let the optimal solution of problem (2) be $\theta^*(\lambda)$ [recall that the optimal solution of problem (1) with parameter λ is denoted by $\beta^*(\lambda)$]. Then, the KKT conditions are given by:

$$\mathbf{y} = \mathbf{X}\beta^*(\lambda) + \lambda\theta^*(\lambda), \tag{3}$$

$$\mathbf{x}_i^T\theta^*(\lambda) \in \begin{cases} \text{sign}([\beta^*(\lambda)]_i), & \text{if } [\beta^*(\lambda)]_i \neq 0, \\ [-1, 1], & \text{if } [\beta^*(\lambda)]_i = 0, \end{cases} \quad i = 1, \ldots, p, \tag{4}$$

where $[\cdot]_k$ denotes the k^{th} component. In view of the KKT condition in (4), we have

$$|\mathbf{x}_i^T(\theta^*(\lambda))^T| < 1 \Rightarrow [\beta^*(\lambda)]_i = 0 \Rightarrow \mathbf{x}_i \text{ is an inactive feature.} \tag{R1}$$

In other words, we can potentially make use of (R1) to identify the inactive features for the Lasso problem. However, since $\theta^*(\lambda)$ is generally unknown, we can not directly apply (R1) to identify the inactive features. Inspired by the SAFE rules (El Ghaoui et al., 2012), we can first estimate a region $\mathbf{\Theta}$ which contains $\theta^*(\lambda')$. Then, (R1) can be relaxed as follows:

$$\sup_{\theta \in \mathbf{\Theta}} |\mathbf{x}_i^T\theta| < 1 \Rightarrow [\beta^*(\lambda)]_i = 0 \Rightarrow \mathbf{x}_i \text{ is an inactive feature.} \tag{R1'}$$

Clearly, as long as we can find a region $\mathbf{\Theta}$ which contains $\theta^*(\lambda)$, (R1') will lead to a screening rule to detect the inactive features for the Lasso problem. Moreover, in view of (R1) and (R1'), we can see that the smaller the region $\mathbf{\Theta}$ is, the more accurate the estimation of $\theta^*(\lambda)$ is. As a result, more inactive features can be identified by the resulting screening rules.

The dual problem has interesting geometric interpretations. By a closer look at the dual problem (2), we can observe that the dual optimal solution is the feasible point which is closest to $\mathbf{y}/\lambda$. For notational convenience, let the feasible set of problem (2) be F. Clearly, F is the intersection of $2p$ closed half-spaces, and thus a closed and convex polytope. (Notice that, F is also nonempty since $0 \in F$.) In other words, $\theta^*(\lambda)$ is the projection of $\mathbf{y}/\lambda$ onto

the polytope F. Mathematically, for an arbitrary vector $\mathbf{w}$ and a convex set C in a Hilbert space $\mathcal{H}$, let us define the projection operator as

$$P_C(\mathbf{w}) = \underset{\mathbf{u}\in C}{\operatorname{argmin}} \|\mathbf{u}-\mathbf{w}\|_2. \tag{5}$$

Then, the dual optimal solution $\theta^*(\lambda)$ can be expressed by

$$\theta^*(\lambda) = P_F(\mathbf{y}/\lambda) = \underset{\theta\in F}{\operatorname{argmin}} \left\|\theta - \frac{\mathbf{y}}{\lambda}\right\|_2. \tag{6}$$

Indeed, the nice property of problem (2) illustrated by (6) leads to many interesting results. For example, it is easy to see that $\mathbf{y}/\lambda$ would be an *interior point* of F when λ is large enough. If this is the case, we immediately have the following assertions: 1) $\mathbf{y}/\lambda$ is an interior point of F implies that none of the constraints of problem (2) would be *active* on $\mathbf{y}/\lambda$, i.e., $|\mathbf{x}_i^T(\mathbf{y}/(\lambda))| < 1$ for all $i = 1,\ldots,p$; 2) $\theta^*(\lambda)$ is an interior point of F as well since $\theta^*(\lambda) = P_F(\mathbf{y}/\lambda) = \mathbf{y}/\lambda$ by (6) and the fact $\mathbf{y}/\lambda \in F$. Combining the results in 1) and 2), it is easy to see that $|\mathbf{x}_i^T\theta^*(\lambda)| < 1$ for all $i = 1,\ldots,p$. By (R1), we can conclude that $\beta^*(\lambda) = 0$, under the assumption that λ is large enough.

The above analysis may naturally lead to a question: does there exist a specific parameter value $\lambda_{\max}$ such that the optimal solution of problem (1) is 0 whenever $\lambda > \lambda_{\max}$? The answer is affirmative. Indeed, let us define

$$\lambda_{\max} = \max_i |\mathbf{x}_i^T\mathbf{y}|. \tag{7}$$

It is well known (Tibshirani et al., 2012) that $\lambda_{\max}$ defined by (7) is the smallest parameter such that problem (1) has a trivial solution, i.e.,

$$\beta^*(\lambda) = 0, \ \ \forall\, \lambda \in [\lambda_{\max}, \infty). \tag{8}$$

Combining the results in (8) and (3), we immediately have

$$\theta^*(\lambda) = \frac{\mathbf{y}}{\lambda}, \ \ \forall\, \lambda \in [\lambda_{\max}, \infty). \tag{9}$$

Therefore, through out the rest of this paper, we will focus on the cases with $\lambda \in (0, \lambda_{\max})$.

In the subsequent sections, we will follow (R1') to develop our screening rules. More specifically, the derivation of the proposed screening rules can be divided into the following three steps:

1. We first estimate a region $\mathbf{\Theta}$ which contains the dual optimal solution $\theta^*(\lambda)$.
2. We solve the maximization problem in (R1'), i.e., $\sup_{\theta\in\mathbf{\Theta}} |\mathbf{x}_i^T\theta|$.
3. By plugging in the upper bound we find in 2, it is straightforward to develop the screening rule based on (R1').

The geometric property of the dual problem illustrated by (6) serves as a fundamentally important role in developing our DPP and EDPP screening rules.

2.2 Fundamental Screening Rules via Dual Polytope Projections (DPP)

In this Section, we propose the so called DPP screening rules for discarding the inactive features for Lasso. As the name indicates, the idea of DPP heavily relies on the properties of projection operators, e.g., the *nonexpansiveness* (Bertsekas, 2003). We will follow the three steps stated in Section 2.1 to develop the DPP screening rules.

First, we need to find a region $\mathbf{\Theta}$ which contains the dual optimal solution $\theta^*(\lambda)$. Indeed, the result in (9) provides us an important clue. That is, we may be able to estimate a possible region for $\theta^*(\lambda)$ in terms of a known $\theta^*(\lambda_0)$ with $\lambda < \lambda_0$. Notice that, we can always set $\lambda_0 = \lambda_{\max}$ and make use of the fact that $\theta^*(\lambda_{\max}) = \mathbf{y}/\lambda_{\max}$ implied by (9). Another key ingredient comes from (6), i.e., the dual optimal solution $\theta^*(\lambda)$ is the projection of $\mathbf{y}/\lambda$ onto the feasible set F, which is nonempty closed and convex. A nice property of the projection operators defined in a Hilbert space with respect to a nonempty closed and convex set is the so called *nonexpansiveness*. For convenience, we restate the definition of nonexpansiveness in the following theorem.

Theorem 1 *Let C be a nonempty closed convex subset of a Hilbert space $\mathcal{H}$. Then the projection operator defined in (5) is continuous and nonexpansive, i.e.,*

$$\|P_C(\mathbf{w}_2) - P_C(\mathbf{w}_1)\|_2 \leq \|\mathbf{w}_2 - \mathbf{w}_1\|_2, \ \forall \mathbf{w}_2, \mathbf{w}_1 \in \mathcal{H}. \tag{10}$$

In view of (6), a direct application of Theorem 1 leads to the following result:

Theorem 2 *For the Lasso problem, let $\lambda, \lambda_0 > 0$ be two regularization parameters. Then,*

$$\|\theta^*(\lambda) - \theta^*(\lambda_0)\|_2 \leq \left|\frac{1}{\lambda} - \frac{1}{\lambda_0}\right| \|\mathbf{y}\|_2. \tag{11}$$

For notational convenience, let a ball centered at $\mathbf{c}$ with radius ρ be denoted by $B(\mathbf{c}, \rho)$. Theorem 2 actually implies that the dual optimal solution must be inside a ball centered at $\theta^*(\lambda_0)$ with radius $|1/\lambda - 1/\lambda_0| \|\mathbf{y}\|_2$, i.e.,

$$\theta^*(\lambda) \in B\left(\theta^*(\lambda_0), \left|\frac{1}{\lambda} - \frac{1}{\lambda_0}\right| \|\mathbf{y}\|_2\right). \tag{12}$$

We thus complete the first step for developing DPP. Because it is easy to find the upper bound of a linear functional over a ball, we combine the remaining two steps as follows.

Theorem 3 *For the Lasso problem, assume that the dual optimum at λ_0, i.e., $\theta^*(\lambda_0)$, is known. Let λ be a positive value different from λ_0. Then $[\beta^*(\lambda)]_i = 0$ if*

$$\left|\mathbf{x}_i^T \theta^*(\lambda)\right| < 1 - \|\mathbf{x}_i\|_2 \|\mathbf{y}\|_2 \left|\frac{1}{\lambda} - \frac{1}{\lambda_0}\right|.$$

Proof The dual optimal solution $\theta^*(\lambda)$ is estimated to be inside the ball given by (12). To simplify notations, let $\mathbf{c} = \theta^*(\lambda_0)$ and $\rho = |1/\lambda - 1/\lambda_0| \|\mathbf{y}\|_2$. To develop a screening rule based on (R1'), we need to solve the optimization problem: $\sup_{\theta \in B(\mathbf{c}, \rho)} |\mathbf{x}_i^T \theta|$.

Indeed, for any $\theta \in B(\mathbf{c}, \rho)$, it can be expressed by:

$$\theta = \theta^*(\lambda_0) + \mathbf{v}, \quad \|\mathbf{v}\|_2 \leq \rho.$$

Therefore, the optimization problem can be easily solved as follows:

$$\sup_{\theta \in B(\mathbf{c},\rho)} \left|\mathbf{x}_i^T \theta\right| = \sup_{\|\mathbf{v}\|_2 \leq \rho} \left|\mathbf{x}_i^T \left(\theta^*(\lambda_0) + \mathbf{v}\right)\right| = \left|\mathbf{x}_i^T \theta^*(\lambda_0)\right| + \rho \|\mathbf{x}_i\|_2. \tag{13}$$

By plugging the upper bound in (13) to (R1'), we obtain the statement in Theorem 3, which completes the proof. ■

Theorem 3 implies that we can develop applicable screening rules for Lasso as long as the dual optimal solution $\theta^*(\cdot)$ is known for a certain parameter value λ_0. By simply setting $\lambda_0 = \lambda_{\max}$ and noting that $\theta^*(\lambda_{\max}) = \mathbf{y}/\lambda_{\max}$ from (9), Theorem 3 immediately leads to the following result.

Corollary 4 Basic DPP: *For the Lasso problem (1), let* $\lambda_{\max} = \max_i |\mathbf{x}_i^T \mathbf{y}|$. *If* $\lambda \geq \lambda_{\max}$, *then* $[\beta^*]_i = 0, \forall i \in \mathcal{I}$. *Otherwise,* $[\beta^*(\lambda)]_i = 0$ *if*

$$\left|\mathbf{x}_i^T \frac{\mathbf{y}}{\lambda_{\max}}\right| < 1 - \left(\frac{1}{\lambda} - \frac{1}{\lambda_{\max}}\right) \|\mathbf{x}_i\|_2 \|\mathbf{y}\|_2.$$

Remark 5 *Notice that, DPP is not the same as ST1 Xiang et al. (2011) and SAFE El Ghaoui et al. (2012), which discards the* i^{th} *feature if*

$$|\mathbf{x}_i^T \mathbf{y}| < \lambda - \|\mathbf{x}_i\|_2 \|\mathbf{y}\|_2 \frac{\lambda_{\max} - \lambda}{\lambda_{\max}}. \tag{14}$$

From the perspective of the sphere test, the radius of ST1/SAFE and DPP are the same. But the centers of ST1 and DPP are $\mathbf{y}/\lambda$ *and* $\mathbf{y}/\lambda_{\max}$ *respectively, which leads to different formulas, i.e., (14) and Corollary 4.*

In real applications, the optimal parameter value of λ is generally unknown and needs to be estimated. To determine an appropriate value of λ, commonly used approaches such as cross validation and stability selection involve solving the Lasso problem over a grid of tuning parameters $\lambda_1 > \lambda_2 > \ldots > \lambda_{\mathcal{K}}$, which is very time consuming. Motivated by the ideas of Tibshirani et al. (2012) and El Ghaoui et al. (2012), we develop a sequential version of DPP rules. We first apply the DPP screening rule in Corollary 4 to discard inactive features for the Lasso problem (1) with parameter being λ_1. After solving the reduced optimization problem for λ_1, we obtain the exact solution $\beta^*(\lambda_1)$. Hence by (3), we can find $\theta^*(\lambda_1)$. According to Theorem 3, once we know the optimal dual solution $\theta^*(\lambda_1)$, we can construct a new screening rule by setting $\lambda_0 = \lambda_1$ to identify inactive features for problem (1) with parameter being λ_2. By repeating the above process, we obtain the sequential version of the DPP rule as in the following corollary.

Corollary 6 Sequential DPP: *For the Lasso problem (1), suppose we are given a sequence of parameter values* $\lambda_{\max} = \lambda_0 > \lambda_1 > \ldots > \lambda_m$. *Then for any integer* $0 \leq k < m$, *we have* $[\beta^*(\lambda_{k+1})]_i = 0$ *if* $\beta^*(\lambda_k)$ *is known and the following holds:*

$$\left|\mathbf{x}_i^T \frac{\mathbf{y} - \mathbf{X}\beta^*(\lambda_k)}{\lambda_k}\right| < 1 - \left(\frac{1}{\lambda_{k+1}} - \frac{1}{\lambda_k}\right) \|\mathbf{x}_i\|_2 \|\mathbf{y}\|_2.$$

Remark 7 *From Corollaries 4 and 6, we can see that both of the DPP and sequential DPP rules discard the inactive features for the Lasso problem with a smaller parameter value by assuming a known dual optimal solution at a larger parameter value. This is in fact a standard way to construct screening rules for Lasso (Tibshirani et al., 2012; El Ghaoui et al., 2012; Xiang et al., 2011).*

Remark 8 *For illustration purpose, we present both the basic and sequential version of the DPP screening rules. However, it is easy to see that the basic DPP rule can be easily derived from its sequential version by simply setting $\lambda_k = \lambda_{\max}$ and $\lambda_{k+1} = \lambda$. Therefore, in this paper, we will focus on the development and evaluation of the sequential version of the proposed screening rules. To avoid any confusions, DPP and EDPP all refer to the corresponding sequential versions.*

2.3 Enhanced DPP Rules for Lasso

In this section, we further improve the DPP rules presented in Section 2.2 by a more careful analysis of the projection operators. Indeed, from the three steps by which we develop the DPP rules, we can see that the first step is a key. In other words, the estimation of the dual optimal solution serves as a fundamentally important role in developing the DPP rules. Moreover, (R1') implies that the more accurate the estimation is, the more effective the resulting screening rule is in discarding the inactive features. The estimation of the dual optimal solution in DPP rules is in fact a direct consequence of the nonexpansiveness of the projection operators. Therefore, in order to improve the performance of the DPP rules in discarding the inactive features, we propose two different approaches to find more accurate estimations of the dual optimal solution. These two approaches are presented in detail in Sections 2.3.1 and 2.3.2 respectively. By combining the ideas of these two approaches, we can further improve the estimation of the dual optimal solution. Based on this estimation, we develop the enhanced DPP rules (EDPP) in Section 2.3.3. Again, we will follow the three steps in Section 2.1 to develop the proposed screening rules.

2.3.1 Improving the DPP rules via Projections of Rays

In the DPP screening rules, the dual optimal solution $\theta^*(\lambda)$ is estimated to be inside the ball $B\left(\theta^*(\lambda_0), |1/\lambda - 1/\lambda_0|\|\mathbf{y}\|_2\right)$ with $\theta^*(\lambda_0)$ given. In this section, we show that $\theta^*(\lambda)$ lies inside a ball centered at $\theta^*(\lambda_0)$ with a smaller radius.

Indeed, it is well known that the projection of an arbitrary point onto a nonempty closed convex set C in a Hilbert space $\mathcal{H}$ always exists and is unique (Bauschke and Combettes, 2011). However, the converse is not true, i.e., there may exist $\mathbf{w}_1, \mathbf{w}_2 \in \mathcal{H}$ such that $\mathbf{w}_1 \neq \mathbf{w}_2$ and $P_C(\mathbf{w}_1) = P_C(\mathbf{w}_2)$. In fact, it is known that the following result holds:

Lemma 9 *(Bauschke and Combettes, 2011) Let C be a nonempty closed convex subset of a Hilbert space $\mathcal{H}$. For a point $\mathbf{w} \in \mathcal{H}$, let $\mathbf{w}(t) = P_C(\mathbf{w}) + t(\mathbf{w} - P_C(\mathbf{w}))$. Then, the projection of the point $\mathbf{w}(t)$ is $P_C(\mathbf{w})$ for all $t \geq 0$, i.e.,*

$$P_C(\mathbf{w}(t)) = P_C(\mathbf{w}), \forall t \geq 0.$$

Clearly, when $\mathbf{w} \neq P_C(\mathbf{w})$, i.e., $\mathbf{w} \notin C$, $\mathbf{w}(t)$ with $t \geq 0$ is the ray starting from $P_C(\mathbf{w})$ and pointing in the same direction as $\mathbf{w} - P_C(\mathbf{w})$. By Lemma 9, we know that the projection

of the ray $\mathbf{w}(t)$ with $t \geq 0$ onto the set C is a single point $P_C(\mathbf{w})$. [When $\mathbf{w} = P_C(\mathbf{w})$, i.e., $\mathbf{w} \in C$, $\mathbf{w}(t)$ with $t \geq 0$ becomes a single point and the statement in Lemma 9 is trivial.]

By making use of Lemma 9 and the nonexpansiveness of the projection operators, we can improve the estimation of the dual optimal solution in DPP [please refer to Theorem 2 and (12)]. More specifically, we have the following result:

Theorem 10 *For the Lasso problem, suppose that the dual optimal solution $\theta^*(\cdot)$ at $\lambda_0 \in (0, \lambda_{\max}]$ is known. For any $\lambda \in (0, \lambda_0]$, let us define*

$$\mathbf{v}_1(\lambda_0) = \begin{cases} \dfrac{\mathbf{y}}{\lambda_0} - \theta^*(\lambda_0), & \text{if } \lambda_0 \in (0, \lambda_{\max}), \\ \text{sign}(\mathbf{x}_*^T \mathbf{y})\mathbf{x}_*, & \text{if } \lambda_0 = \lambda_{\max}, \end{cases} \quad \text{where } \mathbf{x}_* = \text{argmax}_{\mathbf{x}_i} |\mathbf{x}_i^T \mathbf{y}|, \tag{15}$$

$$\mathbf{v}_2(\lambda, \lambda_0) = \frac{\mathbf{y}}{\lambda} - \theta^*(\lambda_0), \tag{16}$$

$$\mathbf{v}_2^\perp(\lambda, \lambda_0) = \mathbf{v}_2(\lambda, \lambda_0) - \frac{\langle \mathbf{v}_1(\lambda_0), \mathbf{v}_2(\lambda, \lambda_0) \rangle}{\|\mathbf{v}_1(\lambda_0)\|_2^2} \mathbf{v}_1(\lambda_0). \tag{17}$$

Then, the dual optimal solution $\theta^*(\lambda)$ can be estimated as follows:

$$\theta^*(\lambda) \in B\left(\theta^*(\lambda_0), \|\mathbf{v}_2^\perp(\lambda, \lambda_0)\|_2\right) \subseteq B\left(\theta^*(\lambda_0), \left|\frac{1}{\lambda} - \frac{1}{\lambda_0}\right| \|\mathbf{y}\|_2\right).$$

Proof By making use of Lemma 9, we present the proof of the statement for the cases with $\lambda_0 \in (0, \lambda_{\max})$. We postpone the proof of the statement for the case with $\lambda_0 = \lambda_{\max}$ after we introduce more general technical results.

In view of the assumption $\lambda_0 \in (0, \lambda_{\max})$, it is easy to see that

$$\frac{\mathbf{y}}{\lambda_0} \notin F \Rightarrow \frac{\mathbf{y}}{\lambda_0} \neq P_F\left(\frac{\mathbf{y}}{\lambda_0}\right) = \theta^*(\lambda_0) \Rightarrow \frac{\mathbf{y}}{\lambda_0} - \theta^*(\lambda_0) \neq 0. \tag{18}$$

For each $\lambda_0 \in (0, \lambda_{\max})$, let us define

$$\theta_{\lambda_0}(t) = \theta^*(\lambda_0) + t\mathbf{v}_1(\lambda_0) = \theta^*(\lambda_0) + t\left(\frac{\mathbf{y}}{\lambda_0} - \theta^*(\lambda_0)\right), \ \ t \geq 0. \tag{19}$$

By the result in (18), we can see that $\theta_{\lambda_0}(\cdot)$ defined by (19) is a ray which starts at $\theta^*(\lambda_0)$ and points in the same direction as $\mathbf{y}/\lambda_0 - \theta^*(\lambda_0)$. In view of (6), a direct application of Lemma 9 leads to that:

$$P_F(\theta_{\lambda_0}(t)) = \theta^*(\lambda_0), \ \ \forall\, t \geq 0. \tag{20}$$

By applying Theorem 1 again, we have

$$\begin{aligned} \|\theta^*(\lambda) - \theta^*(\lambda_0)\|_2 &= \left\| P_F\left(\frac{\mathbf{y}}{\lambda}\right) - P_F(\theta_{\lambda_0}(t)) \right\|_2 \\ &\leq \left\| \frac{\mathbf{y}}{\lambda} - \theta_{\lambda_0}(t) \right\|_2 = \left\| t\left(\frac{\mathbf{y}}{\lambda_0} - \theta^*(\lambda_0)\right) - \left(\frac{\mathbf{y}}{\lambda} - \theta^*(\lambda_0)\right) \right\|_2 \\ &= \|t\mathbf{v}_1(\lambda_0) - \mathbf{v}_2(\lambda, \lambda_0)\|_2, \ \ \forall\, t \geq 0. \end{aligned} \tag{21}$$

Because the inequality in (21) holds for all $t \geq 0$, it is easy to see that

$$\begin{aligned} \|\theta^*(\lambda) - \theta^*(\lambda_0)\|_2 &\leq \min_{t\geq 0} \ \|t\mathbf{v}_1(\lambda_0) - \mathbf{v}_2(\lambda, \lambda_0)\|_2 \\ &= \begin{cases} \|\mathbf{v}_2(\lambda, \lambda_0)\|_2, & \text{if } \langle \mathbf{v}_1(\lambda_0), \mathbf{v}_2(\lambda, \lambda_0)\rangle < 0, \\ \left\|\mathbf{v}_2^{\perp}(\lambda, \lambda_0)\right\|_2, & \text{otherwise.} \end{cases} \end{aligned} \tag{22}$$

The inequality in (22) implies that, to prove the first half of the statement, i.e., $\theta^*(\lambda) \in B(\theta^*(\lambda_0), \|\mathbf{v}_2^{\perp}(\lambda, \lambda_0)\|_2)$, we only need to show that $\langle \mathbf{v}_1(\lambda_0), \mathbf{v}_2(\lambda, \lambda_0)\rangle \geq 0$.

Indeed, it is easy to see that $0 \in F$. Therefore, in view of (20), the distance between $\theta_{\lambda_0}(t)$ and $\theta^*(\lambda_0)$ must be shorter than the one between $\theta_{\lambda_0}(t)$ and 0 for all $t \geq 0$, i.e.,

$$\begin{aligned} &\|\theta_{\lambda_0}(t) - \theta^*(\lambda_0)\|_2^2 \leq \|\theta_{\lambda_0}(t) - 0\|_2^2 \\ \Rightarrow\ &0 \leq \|\theta^*(\lambda_0)\|_2^2 + 2t\left(\left\langle \theta^*(\lambda_0), \frac{\mathbf{y}}{\lambda_0}\right\rangle - \|\theta^*(\lambda_0)\|_2^2\right), \ \ \forall\, t \geq 0. \end{aligned} \tag{23}$$

Since the inequality in (23) holds for all $t \geq 0$, we can conclude that:

$$\left\langle \theta^*(\lambda_0), \frac{\mathbf{y}}{\lambda_0}\right\rangle - \|\theta^*(\lambda_0)\|_2^2 \geq 0 \Rightarrow \frac{\|\mathbf{y}\|_2}{\lambda_0} \geq \|\theta^*(\lambda_0)\|_2. \tag{24}$$

Therefore, we can see that:

$$\begin{aligned} \langle \mathbf{v}_1(\lambda_0), \mathbf{v}_2(\lambda, \lambda_0)\rangle &= \left\langle \frac{\mathbf{y}}{\lambda_0} - \theta^*(\lambda_0), \frac{\mathbf{y}}{\lambda} - \frac{\mathbf{y}}{\lambda_0} + \frac{\mathbf{y}}{\lambda_0} - \theta^*(\lambda_0)\right\rangle \\ &\geq \left(\frac{1}{\lambda} - \frac{1}{\lambda_0}\right)\left\langle \frac{\mathbf{y}}{\lambda_0} - \theta^*(\lambda_0), \mathbf{y}\right\rangle \\ &= \left(\frac{1}{\lambda} - \frac{1}{\lambda_0}\right)\left(\frac{\|\mathbf{y}\|_2^2}{\lambda_0} - \langle \theta^*(\lambda_0), \mathbf{y}\rangle\right) \\ &\geq \left(\frac{1}{\lambda} - \frac{1}{\lambda_0}\right)\left(\frac{\|\mathbf{y}\|_2^2}{\lambda_0} - \|\theta^*(\lambda_0)\|_2\|\mathbf{y}\|_2\right) \geq 0. \end{aligned} \tag{25}$$

The last inequality in (25) is due to the result in (24).

Clearly, in view of (22) and (25), we can see that the first half of the statement holds, i.e., $\theta^*(\lambda) \in B(\theta^*(\lambda_0), \|\mathbf{v}_2^{\perp}(\lambda, \lambda_0)\|_2)$. The second half of the statement, i.e., $B(\theta^*(\lambda_0), \|\mathbf{v}_2^{\perp}(\lambda, \lambda_0)\|_2) \subseteq B(\theta^*(\lambda_0), |1/\lambda - 1/\lambda_0|\|\mathbf{y}\|_2)$, can be easily obtained by noting that the inequality in (21) reduces to the one in (12) when $t = 1$. This completes the proof of the statement with $\lambda_0 \in (0, \lambda_{\max})$. ■

Before we present the proof of Theorem 10 for the case with $\lambda_0 = \lambda_{\max}$, let us briefly review some technical results from convex analysis first.

Definition 11 *(Ruszczyński, 2006) Let C be a nonempty closed convex subset of a Hilbert space $\mathcal{H}$ and $\mathbf{w} \in C$. The set*

$$N_C(\mathbf{w}) := \{\mathbf{v} : \langle \mathbf{v}, \mathbf{u} - \mathbf{w}\rangle \leq 0, \forall \mathbf{u} \in C\}$$

is called the normal cone to C at $\mathbf{w}$.

In terms of the normal cones, the following theorem provides an elegant and useful characterization of the projections onto nonempty closed convex subsets of a Hilbert space.

Theorem 12 *(Bauschke and Combettes, 2011) Let C be a nonempty closed convex subset of a Hilbert space $\mathcal{H}$. Then, for every $\mathbf{w} \in \mathcal{H}$ and $\mathbf{w}_0 \in C$, $\mathbf{w}_0$ is the projection of $\mathbf{w}$ onto C if and only if $\mathbf{w} - \mathbf{w}_0 \in N_C(\mathbf{w}_0)$, i.e.,*

$$\mathbf{w}_0 = P_C(\mathbf{w}) \Leftrightarrow \langle \mathbf{w} - \mathbf{w}_0, \mathbf{u} - \mathbf{w}_0 \rangle \leq 0, \forall \mathbf{u} \in C.$$

In view of the proof of Theorem 10, we can see that (20) is a key step. When $\lambda_0 = \lambda_{\max}$, similar to (19), let us define

$$\theta_{\lambda_{\max}}(t) = \theta^*(\lambda_{\max}) + t\mathbf{v}_1(\lambda_{\max}), \ \ \forall\, t \geq 0. \tag{26}$$

By Theorem 12, the following lemma shows that (20) also holds for $\lambda_0 = \lambda_{\max}$.

Lemma 13 *For the Lasso problem, let $\mathbf{v}_1(\cdot)$ and $\theta_{\lambda_{\max}}(\cdot)$ be given by (15) and (26), then the following result holds:*

$$P_F(\theta_{\lambda_{\max}}(t)) = \theta^*(\lambda_{\max}), \ \forall\ t \geq 0. \tag{27}$$

Proof To prove the statement, Theorem 12 implies that we only need to show:

$$\langle \mathbf{v}_1(\lambda_{\max}), \theta - \theta^*(\lambda_{\max}) \rangle \leq 0, \ \ \forall\, \theta \in F. \tag{28}$$

Recall that $\mathbf{v}_1(\lambda_{\max}) = \text{sign}(\mathbf{x}_*^T\mathbf{y})\mathbf{x}_*$, $\mathbf{x}_* = \text{argmax}_{\mathbf{x}_i} |\mathbf{x}_i^T\mathbf{y}|$ from (15), and $\theta^*(\lambda_{\max}) = \mathbf{y}/\lambda_{\max}$ from (9). It is easy to see that

$$\langle \mathbf{v}_1(\lambda_{\max}), \theta^*(\lambda_{\max}) \rangle = \left\langle \text{sign}(\mathbf{x}_*^T\mathbf{y})\mathbf{x}_*, \frac{\mathbf{y}}{\lambda_{\max}} \right\rangle = \frac{|\mathbf{x}_*^T\mathbf{y}|}{\lambda_{\max}} = 1. \tag{29}$$

Moreover, assume θ is an arbitrary point of F. Then, we have $|\langle \mathbf{x}_*, \theta \rangle| \leq 1$, and thus

$$\langle \mathbf{v}_1(\lambda_{\max}), \theta \rangle = \langle \text{sign}(\mathbf{x}_*^T\mathbf{y})\mathbf{x}_*, \theta \rangle \leq |\langle \mathbf{x}_*, \theta \rangle| \leq 1. \tag{30}$$

Therefore, the inequality in (28) easily follows by combing the results in (29) and (30), which completes the proof. ■

We are now ready to give the proof of Theorem 10 for the case with $\lambda_0 = \lambda_{\max}$.

Proof In view of Theorem 1 and Lemma 13, we have

$$\begin{aligned} \|\theta^*(\lambda) - \theta^*(\lambda_{\max})\|_2 &= \left\| P_F\left(\frac{\mathbf{y}}{\lambda}\right) - P_F(\theta_{\lambda_{\max}}(t)) \right\|_2 \\ &\leq \left\| \frac{\mathbf{y}}{\lambda} - \theta_{\lambda_{\max}}(t) \right\|_2 = \left\| t\mathbf{v}_1(\lambda_{\max}) - \left(\frac{\mathbf{y}}{\lambda} - \theta^*(\lambda_{\max})\right) \right\|_2 \\ &= \|t\mathbf{v}_1(\lambda_{\max}) - \mathbf{v}_2(\lambda, \lambda_{\max})\|_2, \ \ \forall\, t \geq 0. \end{aligned} \tag{31}$$

Because the inequality in (31) holds for all $t \geq 0$, we can see that

$$\|\theta^*(\lambda) - \theta^*(\lambda_{\max})\|_2 \leq \min_{t\geq 0} \|t\mathbf{v}_1(\lambda_{\max}) - \mathbf{v}_2(\lambda, \lambda_{\max})\|_2 \tag{32}$$
$$= \begin{cases} \|\mathbf{v}_2(\lambda, \lambda_{\max})\|_2, & \text{if } \langle \mathbf{v}_1(\lambda_{\max}), \mathbf{v}_2(\lambda, \lambda_{\max})\rangle < 0, \\ \left\|\mathbf{v}_2^{\perp}(\lambda, \lambda_{\max})\right\|_2, & \text{otherwise.} \end{cases}$$

Clearly, we only need to show that $\langle \mathbf{v}_1(\lambda_{\max}), \mathbf{v}_2(\lambda, \lambda_{\max})\rangle \geq 0$.

Indeed, Lemma 13 implies that $\mathbf{v}_1(\lambda_{\max}) \in N_F(\theta^*(\lambda_{\max}))$ [please refer to the inequality in (28)]. By noting that $0 \in F$, we have

$$\left\langle \mathbf{v}_1(\lambda_{\max}), 0 - \frac{\mathbf{y}}{\lambda_{\max}} \right\rangle \leq 0 \Rightarrow \langle \mathbf{v}_1(\lambda_{\max}), \mathbf{y}\rangle \geq 0.$$

Moreover, because $\mathbf{y}/\lambda_{\max} = \theta^*(\lambda_{\max})$, it is easy to see that

$$\langle \mathbf{v}_1(\lambda_{\max}), \mathbf{v}_2(\lambda, \lambda_{\max})\rangle = \left\langle \mathbf{v}_1(\lambda_{\max}), \frac{\mathbf{y}}{\lambda} - \frac{\mathbf{y}}{\lambda_{\max}} \right\rangle \tag{33}$$
$$= \left(\frac{1}{\lambda} - \frac{1}{\lambda_{\max}}\right) \langle \mathbf{v}_1(\lambda_{\max}), \mathbf{y}\rangle \geq 0.$$

Therefore, in view of (32) and (33), we can see that the first half of the statement holds, i.e., $\theta^*(\lambda) \in B(\theta^*(\lambda_{\max}), \|\mathbf{v}_2^{\perp}(\lambda, \lambda_{\max})\|_2)$. The second half of the statement, i.e., $B(\theta^*(\lambda_{\max}), \|\mathbf{v}_2^{\perp}(\lambda, \lambda_{\max})\|_2) \subseteq B(\theta^*(\lambda_{\max}), |1/\lambda - 1/\lambda_{\max}|\|\mathbf{y}\|_2)$, can be easily obtained by noting that the inequality in (32) reduces to the one in (12) when $t = 0$. This completes the proof of the statement with $\lambda_0 = \lambda_{\max}$. Thus, the proof of Theorem 10 is completed. ■

Theorem 10 in fact provides a more accurate estimation of the dual optimal solution than the one in DPP, i.e., $\theta^*(\lambda)$ lies inside a ball centered at $\theta^*(\lambda_0)$ with a radius $\|\mathbf{v}_2^{\perp}(\lambda, \lambda_0)\|_2$. Based on this improved estimation and (R1'), we can develop the following screening rule to discard the inactive features for Lasso.

Theorem 14 *For the Lasso problem, assume the dual optimal solution $\theta^*(\cdot)$ at $\lambda_0 \in (0, \lambda_{\max}]$ is known. Then, for each $\lambda \in (0, \lambda_0)$, we have $[\beta^*(\lambda)]_i = 0$ if*

$$|\mathbf{x}_i^T \theta^*(\lambda_0)| < 1 - \|\mathbf{v}_2^{\perp}(\lambda, \lambda_0)\|_2 \|\mathbf{x}_i\|_2.$$

We omit the proof of Theorem 14 since it is very similar to the one of Theorem 3. By Theorem 14, we can easily develop the following sequential screening rule.

Improvement 1: *For the Lasso problem (1), suppose we are given a sequence of parameter values $\lambda_{\max} = \lambda_0 > \lambda_1 > \ldots > \lambda_{\mathcal{K}}$. Then for any integer $0 \leq k < \mathcal{K}$, we have $[\beta^*(\lambda_{k+1})]_i = 0$ if $\beta^*(\lambda_k)$ is known and the following holds:*

$$\left|\mathbf{x}_i^T \frac{\mathbf{y} - \mathbf{X}\beta^*(\lambda_k)}{\lambda_k}\right| < 1 - \|\mathbf{v}_2^{\perp}(\lambda_{k+1}, \lambda_k)\|_2 \|\mathbf{x}_i\|_2.$$

The screening rule in Improvement 1 is developed based on (R1') and the estimation of the dual optimal solution in Theorem 10, which is more accurate than the one in DPP. Therefore, in view of (R1'), the screening rule in Improvement 1 are more effective in discarding the inactive features than the DPP rule.

2.3.2 Improving the DPP rules via Firmly Nonexpansiveness

In Section 2.3.1, we improve the estimation of the dual optimal solution in DPP by making use of the projections of properly chosen rays. (R1') implies that the resulting screening rule stated in Improvement 1 is more effective in discarding the inactive features than DPP. In this Section, we present another approach to improve the estimation of the dual optimal solution in DPP by making use of the so called *firmly nonexpansiveness* of the projections onto nonempty closed convex subset of a Hilbert space.

Theorem 15 *(Bauschke and Combettes, 2011) Let C be a nonempty closed convex subset of a Hilbert space $\mathcal{H}$. Then the projection operator defined in (5) is continuous and firmly nonexpansive. In other words, for any $\mathbf{w}_1, \mathbf{w}_2 \in \mathcal{H}$, we have*

$$\|P_C(\mathbf{w}_1) - P_C(\mathbf{w}_2)\|_2^2 + \|(\mathrm{Id} - P_C)(\mathbf{w}_1) - (\mathrm{Id} - P_C)(\mathbf{w}_2)\|_2^2 \leq \|\mathbf{w}_1 - \mathbf{w}_2\|_2^2, \tag{34}$$

where Id *is the identity operator.*

In view of the inequalities in (34) and (10), it is easy to see that firmly nonexpansiveness implies nonexpansiveness. But the converse is not true. Therefore, firmly nonexpansiveness of the projection operators is a stronger property than the nonexpansiveness. A direct application of Theorem 15 leads to the following result.

Theorem 16 *For the Lasso problem, let $\lambda, \lambda_0 > 0$ be two parameter values. Then*

$$\theta^*(\lambda) \in B\left(\theta^*(\lambda_0) + \frac{1}{2}\left(\frac{1}{\lambda} - \frac{1}{\lambda_0}\right)\mathbf{y}, \frac{1}{2}\left|\frac{1}{\lambda} - \frac{1}{\lambda_0}\right| \|\mathbf{y}\|_2\right) \subset B\left(\theta^*(\lambda_0), \left|\frac{1}{\lambda} - \frac{1}{\lambda_0}\right| \|\mathbf{y}\|_2\right). \tag{35}$$

Proof In view of (6) and the firmly nonexpansiveness in (34), we have

$$\begin{aligned} & \|\theta^*(\lambda) - \theta^*(\lambda_0)\|_2^2 + \left\|\left(\frac{\mathbf{y}}{\lambda} - \theta^*(\lambda)\right) - \left(\frac{\mathbf{y}}{\lambda_0} - \theta^*(\lambda_0)\right)\right\|_2^2 \leq \left\|\frac{\mathbf{y}}{\lambda} - \frac{\mathbf{y}}{\lambda_0}\right\|_2^2 \qquad (36) \\ \Leftrightarrow\ & \|\theta^*(\lambda) - \theta^*(\lambda_0)\|_2^2 \leq \left\langle \theta^*(\lambda) - \theta^*(\lambda_0), \frac{\mathbf{y}}{\lambda} - \frac{\mathbf{y}}{\lambda_0} \right\rangle \\ \Leftrightarrow\ & \left\|\theta^*(\lambda) - \left(\theta^*(\lambda_0) + \frac{1}{2}\left(\frac{1}{\lambda} - \frac{1}{\lambda_0}\right)\mathbf{y}\right)\right\|_2 \leq \frac{1}{2}\left|\frac{1}{\lambda} - \frac{1}{\lambda_0}\right| \|\mathbf{y}\|_2, \end{aligned}$$

which completes the proof of the first half of the statement. The second half of the statement is trivial by noting that the first inequality in (36) (firmly nonexpansiveness) implies the inequality in (11) (nonexpansiveness) but not vice versa. Indeed, it is easy to see that the ball in the middle of (35) is inside the right one and has only a half radius. ■

Clearly, Theorem 16 provides a more accurate estimation of the dual optimal solution than the one in DPP, i.e., the dual optimal solution must be inside a ball which is a subset of the one in DPP and has only a half radius. Again, based on the estimation in Theorem 16 and (R1'), we have the following result.

Theorem 17 *For the Lasso problem, assume that the dual optimal solution $\theta^*(\cdot)$ at $\lambda_0 \in (0, \lambda_{\max}]$ is known. Then, for each $\lambda \in (0, \lambda_0)$, we have $[\beta^*(\lambda)]_i = 0$ if*

$$\left| \mathbf{x}_i^T \left(\theta^*(\lambda_0) + \frac{1}{2}\left(\frac{1}{\lambda} - \frac{1}{\lambda_0}\right)\mathbf{y} \right) \right| < 1 - \frac{1}{2}\left(\frac{1}{\lambda} - \frac{1}{\lambda_0}\right)\|\mathbf{y}\|_2\|\mathbf{x}_i\|_2.$$

We omit the proof of Theorem 17 since it is very similar to the proof of Theorem 3. A direct application of Theorem 17 leads to the following sequential screening rule.

Improvement 2: *For the Lasso problem (1), suppose that we are given a sequence of parameter values $\lambda_{\max} = \lambda_0 > \lambda_1 > \ldots > \lambda_{\mathcal{K}}$. Then for any integer $0 \leq k < \mathcal{K}$, we have $[\beta^*(\lambda_{k+1})]_i = 0$ if $\beta^*(\lambda_k)$ is known and the following holds:*

$$\left| \mathbf{x}_i^T \left(\frac{\mathbf{y} - \mathbf{X}\beta^*(\lambda_k)}{\lambda_k} + \frac{1}{2}\left(\frac{1}{\lambda_{k+1}} - \frac{1}{\lambda_k}\right)\mathbf{y} \right) \right| < 1 - \frac{1}{2}\left(\frac{1}{\lambda_{k+1}} - \frac{1}{\lambda_k}\right)\|\mathbf{y}\|_2\|\mathbf{x}_i\|_2.$$

Because the screening rule in Improvement 2 is developed based on (R1') and the estimation in Theorem 16, it is easy to see that Improvement 2 is more effective in discarding the inactive features than DPP.

2.3.3 The Proposed Enhanced DPP Rules

In Sections 2.3.1 and 2.3.2, we present two different approaches to improve the estimation of the dual optimal solution in DPP. In view of (R1'), we can see that the resulting screening rules, i.e., Improvements 1 and 2, are more effective in discarding the inactive features than DPP. In this section, we give a more accurate estimation of the dual optimal solution than the ones in Theorems 10 and 16 by combining the aforementioned two approaches together. The resulting screening rule for Lasso is the so called enhanced DPP rule (EDPP). Again, (R1') implies that EDPP is more effective in discarding the inactive features than the screening rules in Improvements 1 and 2. We also present several experiments to demonstrate that EDPP is able to identify more inactive features than the screening rules in Improvements 1 and 2. Therefore, in the subsequent sections, we will focus on the generalizations and evaluations of EDPP.

To develop the EDPP rules, we still follow the three steps in Section 2.1. Indeed, by combining the two approaches proposed in Sections 2.3.1 and 2.3.2, we can further improve the estimation of the dual optimal solution in the following theorem.

Theorem 18 *For the Lasso problem, suppose that the dual optimal solution $\theta^*(\cdot)$ at $\lambda_0 \in (0, \lambda_{\max}]$ is known, and $\forall\, \lambda \in (0, \lambda_0]$, let $\mathbf{v}_2^\perp(\lambda, \lambda_0)$ be given by (17). Then, we have*

$$\left\| \theta^*(\lambda) - \left(\theta^*(\lambda_0) + \frac{1}{2}\mathbf{v}_2^\perp(\lambda, \lambda_0) \right) \right\|_2 \leq \frac{1}{2}\|\mathbf{v}_2^\perp(\lambda, \lambda_0)\|_2.$$

Proof Recall that $\theta_{\lambda_0}(t)$ is defined by (19) and (26). In view of (34), we have

$$\left\| P_F\left(\frac{\mathbf{y}}{\lambda}\right) - P_F\left(\theta_{\lambda_0}(t)\right) \right\|_2^2 + \left\| (\mathrm{Id} - P_F)\left(\frac{\mathbf{y}}{\lambda}\right) - (\mathrm{Id} - P_F)\left(\theta_{\lambda_0}(t)\right) \right\|_2^2 \leq \left\| \frac{\mathbf{y}}{\lambda} - \theta_{\lambda_0}(t) \right\|_2^2. \tag{37}$$

By expanding the second term on the left hand side of (37) and rearranging the terms, we obtain the following equivalent form:

$$\left\| P_F\left(\frac{\mathbf{y}}{\lambda}\right) - P_F\left(\theta_{\lambda_0}(t)\right)\right\|_2^2 \leq \left\langle \frac{\mathbf{y}}{\lambda} - \theta_{\lambda_0}(t), P_F\left(\frac{\mathbf{y}}{\lambda}\right) - P_F\left(\theta_{\lambda_0}(t)\right)\right\rangle. \tag{38}$$

In view of (6), (20) and (27), the inequality in (38) can be rewritten as

$$\begin{aligned} \|\theta^*(\lambda) - \theta^*(\lambda_0)\|_2^2 &\leq \left\langle \frac{\mathbf{y}}{\lambda} - \theta_{\lambda_0}(t), \theta^*(\lambda) - \theta^*(\lambda_0)\right\rangle \\ &= \left\langle \frac{\mathbf{y}}{\lambda} - \theta^*(\lambda_0) - t\mathbf{v}_1(\lambda_0), \theta^*(\lambda) - \theta^*(\lambda_0)\right\rangle \\ &= \langle \mathbf{v}_2(\lambda,\lambda_0) - t\mathbf{v}_1(\lambda_0), \theta^*(\lambda) - \theta^*(\lambda_0)\rangle, \ \ \forall t \geq 0. \end{aligned} \tag{39}$$

[Recall that $\mathbf{v}_1(\lambda_0)$ and $\mathbf{v}_2(\lambda,\lambda_0)$ are defined by (15) and (16) respectively.] Clearly, the inequality in (39) is equivalent to

$$\left\|\theta^*(\lambda) - \left(\theta^*(\lambda_0) + \frac{1}{2}(\mathbf{v}_2(\lambda,\lambda_0) - t\mathbf{v}_1(\lambda_0))\right)\right\|_2^2 \leq \frac{1}{4}\|\mathbf{v}_2(\lambda,\lambda_0) - t\mathbf{v}_1(\lambda_0)\|_2^2, \ \ \forall t \geq 0. \tag{40}$$

The statement follows easily by minimizing the right hand side of the inequality in (40), which has been done in the proof of Theorem 10. ■

Indeed, Theorem 18 is equivalent to bounding $\theta^*(\lambda)$ in a ball as follows:

$$\theta^*(\lambda) \in B\left(\theta^*(\lambda_0) + \frac{1}{2}\mathbf{v}_2^\perp(\lambda,\lambda_0), \frac{1}{2}\|\mathbf{v}_2^\perp(\lambda,\lambda_0)\|_2\right). \tag{41}$$

Based on this estimation and (R1'), we immediately have the following result.

Theorem 19 *For the Lasso problem, assume that the dual optimal problem $\theta^*(\cdot)$ at $\lambda_0 \in (0, \lambda_{\max}]$ is known, and $\lambda \in (0, \lambda_0]$. Then $[\beta^*(\lambda)]_i = 0$ if the following holds:*

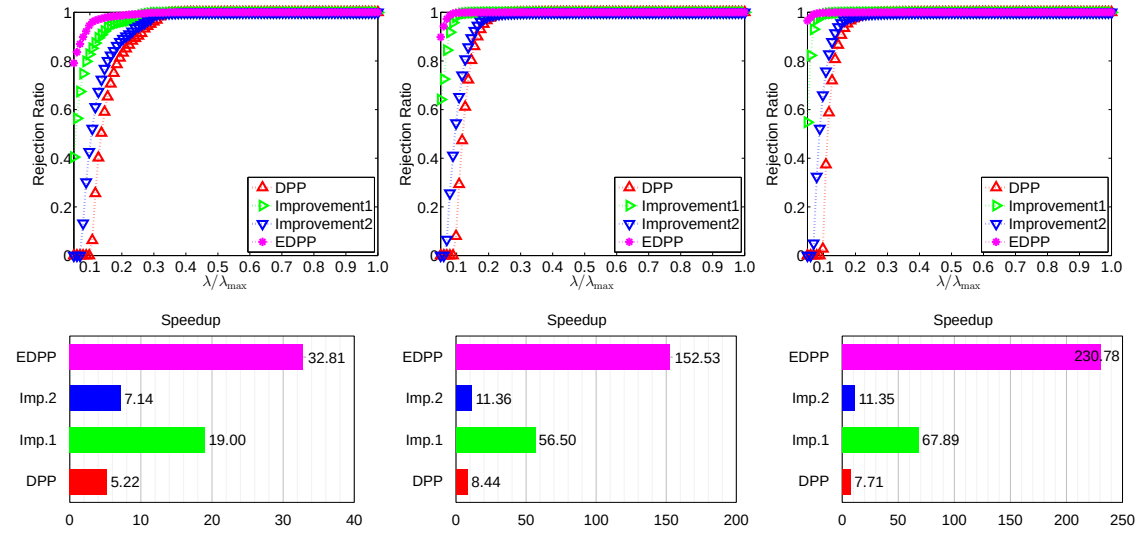
$$\left|\mathbf{x}_i^T\left(\theta^*(\lambda_0) + \frac{1}{2}\mathbf{v}_2^\perp(\lambda,\lambda_0)\right)\right| < 1 - \frac{1}{2}\|\mathbf{v}_2^\perp(\lambda,\lambda_0)\|_2\|\mathbf{x}_i\|_2.$$

We omit the proof of Theorem 19 since it is very similar to the one of Theorem 3. Based on Theorem 19, we can develop the EDPP rules as follows.

Corollary 20 EDPP: *For the Lasso problem, suppose that we are given a sequence of parameter values $\lambda_{\max} = \lambda_0 > \lambda_1 > \ldots > \lambda_{\mathcal{K}}$. Then for any integer $0 \leq k < \mathcal{K}$, we have $[\beta^*(\lambda_{k+1})]_i = 0$ if $\beta^*(\lambda_k)$ is known and the following holds:*

$$\left|\mathbf{x}_i^T\left(\frac{\mathbf{y} - \mathbf{X}\beta^*(\lambda_k)}{\lambda_k} + \frac{1}{2}\mathbf{v}_2^\perp(\lambda_{k+1},\lambda_k)\right)\right| < 1 - \frac{1}{2}\|\mathbf{v}_2^\perp(\lambda_{k+1},\lambda_k)\|_2\|\mathbf{x}_i\|_2. \tag{42}$$

It is easy to see that the ball in (41) has the smallest radius compared to the ones in Theorems 10 and 16, and thus it provides the most accurate estimation of the dual optimal solution. According to (R1'), EDPP is more effective in discarding the inactive features than DPP, Improvements 1 and 2.



(a) Prostate Cancer, $\mathbf{X} \in \mathbb{R}^{132\times 15154}$ (b) PIE, $\mathbf{X} \in \mathbb{R}^{1024\times 11553}$ (c) MNIST, $\mathbf{X} \in \mathbb{R}^{784\times 50000}$

Figure 1: Comparison of the family of DPP rules on three real data sets: Prostate Cancer digit data set (left), PIE data set (middle) and MNIST image data set (right). The first row shows the rejection ratios of DPP, Improvement 1, Improvement 2 and EDPP. The second row presents the speedup gained by these four methods.

We evaluate the performance of the family of DPP screening rules, i.e., DPP, Improvement 1, Improvement 2 and EDPP, on three real data sets: a) the Prostate Cancer (Petricoin et al., 2002); b) the PIE face image data set (Sim et al., 2003); c) the MNIST handwritten digit data set (Lecun et al., 1998). To measure the performance of the screening rules, we compute the following two quantities:

1. the *rejection ratio*, i.e., the ratio of the number of features discarded by screening rules to the actual number of zero features in the ground truth;

2. the *speedup*, i.e., the ratio of the running time of the solver with screening rules to the running time of the solver without screening.

For each data set, we run the solver with or without the screening rules to solve the Lasso problem along a sequence of 100 parameter values equally spaced on the $\lambda/\lambda_{\max}$ scale from 0.05 to 1.0. Figure 1 presents the rejection ratios and speedup by the family of DPP screening rules. Table 1 reports the running time of the solver with or without the screening rules for solving the 100 Lasso problems, as well as the time for running the screening rules.

The Prostate Cancer data set (Petricoin et al., 2002) is obtained by protein mass spectrometry. The features are indexed by time-of-flight values, which are related to the mass over charge ratios of the constituent proteins in the blood. The data set has 15154 measurements of 132 patients. 69 of the patients have prostate cancer and the rest are healthy. Therefore, the data matrix $\mathbf{X}$ is of size 132×15154, and the response vector $\mathbf{y} \in \{1, -1\}^{132}$ contains the binary labels of the patients.

Data	solver	DPP+solver	Imp.1+solver	Imp.2+solver	EDPP+solver	DPP	Imp.1	Imp.2	EDPP
Prostate Cancer	121.41	23.36	6.39	17.00	3.70	0.30	0.27	0.28	0.23
PIE	629.94	74.66	11.15	55.45	4.13	1.63	1.34	1.54	1.33
MNIST	2566.26	332.87	37.80	226.02	11.12	5.28	4.36	4.94	4.19

Table 1: Running time (in seconds) for solving the Lasso problems along a sequence of 100 tuning parameter values equally spaced on the scale of $\lambda/\lambda_{\max}$ from 0.05 to 1 by (a): the solver (Liu et al., 2009) (reported in the second column) without screening; (b): the solver combined with different screening methods (reported in the 3^{rd} to the 6^{th} columns). The last four columns report the total running time (in seconds) for the screening methods.

The PIE face image data set used in this experiment[1] (Cai et al., 2007) contains 11554 gray face images of 68 people, taken under different poses, illumination conditions and expressions. Each of the images has 32×32 pixels. Therefore, in each trial, we first randomly pick an image as the response $\mathbf{y} \in \mathbb{R}^{1024}$, and then use the remaining images to form the data matrix $\mathbf{X} \in \mathbb{R}^{1024 \times 11553}$. We run 100 trials and report the average performance of the screening rules.

The MNIST data set contains gray images of scanned handwritten digits, including $60,000$ for training and $10,000$ for testing. The dimension of each image is 28×28. We first randomly select 5000 images for each digit from the training set (and in total we have 50000 images) and get a data matrix $\mathbf{X} \in \mathbb{R}^{784 \times 50000}$. Then in each trial, we randomly select an image from the testing set as the response $\mathbf{y} \in \mathbb{R}^{784}$. We run 100 trials and report the average performance of the screening rules.

From Figure 1, we can see that both Improvements 1 and 2 are able to discard more inactive features than DPP, and thus lead to a higher speedup. Compared to Improvement 2, we can also observe that Improvement 1 is more effective in discarding the inactive features. For the three data sets, the second row of Figure 1 shows that Improvement 1 leads to about 20, 60, 70 times speedup respectively, which are much higher than the ones gained by Improvement 1 (roughly 10 times for all the three cases).

Moreover, the EDPP rule, which combines the ideas of both Improvements 1 and 2, is even more effective in discarding the inactive features than Improvement 1. We can see that, for all of the three data sets and most of the 100 parameter values, the rejection ratios of EDPP are very close to 100%. In other words, EDPP is able to discard almost all of the inactive features. Thus, the resulting speedup of EDPP is significantly better than the ones gained by the other three DPP rules. For the PIE and MNIST data sets, we can see that the speedup gained EDPP is about 150 and 230 times, which are two orders of magnitude. In view of Table 1, for the MNIST data set, the solver without screening needs about 2566.26 seconds to solve the 100 Lasso problems. In contrast, the solver with EDPP only needs 11.12 seconds, leading to substantial savings in the computational cost. Moreover, from

1. `http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html`

the last four columns of Table 1, we can also observe that the computational cost of the family of DPP rules are very low. Compared to that of the solver without screening, the computational cost of the family of DPP rules is negligible.

In Section 4, we will only compare the performance of EDPP against several other state-of-the-art screening rules.

3. Extensions to Group Lasso

To demonstrate the flexibility of the family of DPP rules, we extend the idea of EDPP to the group Lasso problem (Yuan and Lin, 2006) in this section. Although the Lasso and group Lasso problems are very different from each other, we will see that their dual problems share a lot of similarities. For example, both of the dual problems can be formulated as looking for projections onto nonempty closed convex subsets of a Hilbert space. Recall that, the EDPP rule for the Lasso problem is entirely based on the properties of the projection operators. Therefore, the framework of the EDPP screening rule we developed for Lasso is also applicable for the group Lasso problem. In Section 3.1, we briefly review some basics of the group Lasso problem and explore the geometric properties of its dual problem. In Section 3.2, we develop the EDPP rule for the group Lasso problem.

3.1 Basics

With the group information available, the group Lasso problem takes the form of:

$$\inf_{\beta\in\mathbb{R}^p} \frac{1}{2}\left\|\mathbf{y}-\sum_{g=1}^{G}\mathbf{X}_g\beta_g\right\|_2^2+\lambda\sum_{g=1}^{G}\sqrt{n_g}\|\beta_g\|_2, \tag{43}$$

where $\mathbf{X}_g\in\mathbb{R}^{N\times n_g}$ is the data matrix for the g^{th} group and $p=\sum_{g=1}^{G}n_g$. The dual problem of (43) is (see detailed derivation in the appendix):

$$\sup_{\theta}\quad\left\{\frac{1}{2}\|\mathbf{y}\|_2^2-\frac{\lambda^2}{2}\left\|\theta-\frac{\mathbf{y}}{\lambda}\right\|_2^2:\ \|\mathbf{X}_g^T\theta\|_2\le\sqrt{n_g},\ g=1,2,\ldots,G\right\} \tag{44}$$

The KKT conditions are given by

$$\mathbf{y}=\sum_{g=1}^{G}\mathbf{X}_g\beta_g^*(\lambda)+\lambda\theta^*(\lambda), \tag{45}$$

$$(\theta^*(\lambda))^T\mathbf{X}_g\in\begin{cases}\sqrt{n_g}\frac{\beta_g^*(\lambda)}{\|\beta_g^*(\lambda)\|_2}, & \text{if}\beta_g^*(\lambda)\neq 0,\\ \sqrt{n_g}\mathbf{u},\ \|\mathbf{u}\|_2\le 1, & \text{if}\beta_g^*(\lambda)=0.\end{cases} \tag{46}$$

for $g=1,2,\ldots,G$. Clearly, in view of (46), we can see that

$$\|(\theta^*(\lambda))^T\mathbf{X}_g\|_2<\sqrt{n_g}\Rightarrow\beta_g^*(\lambda)=0 \tag{R2}$$

However, since $\theta^*(\lambda)$ is generally unknown, (R2) is not applicable to identify the *inactive groups*, i.e., the groups which have 0 coefficients in the solution vector, for the group Lasso problem. Therefore, similar to the Lasso problem, we can first find a region $\overline{\mathbf{\Theta}}$ which contains $\theta^*(\lambda)$, and then (R2) can be relaxed as follows:

$$\sup_{\theta\in\overline{\mathbf{\Theta}}}\|(\theta)^T\mathbf{X}_g\|_2<\sqrt{n_g}\Rightarrow\beta_g^*(\lambda)=0. \tag{R2$'$}$$

Therefore, to develop screening rules for the group Lasso problem, we only need to estimate the region $\overline{\mathbf{\Theta}}$ which contains $\theta^*(\lambda)$, solve the maximization problem in (R2′), and plug it into (R2′). In other words, the three steps proposed in Section 2.1 can also be applied to develop screening rules for the group Lasso problem. Moreover, (R2′) also implies that the smaller the region $\overline{\mathbf{\Theta}}$ is, the more accurate the estimation of the dual optimal solution is. As a result, the more effective the resulting screening rule is in discarding the inactive features.

The dual problem of group Lasso has similar geometric interpretations to the one of Lasso. For notational convenience, let $\overline{F}$ be the feasible set of problem (44). Similar to the case of Lasso, problem (44)implies that the dual optimal $\theta^*(\lambda)$ is the projection of $\mathbf{y}/\lambda$ onto the feasible set $\overline{F}$, i.e.,

$$\theta^*(\lambda) = P_{\overline{F}}\left(\frac{\mathbf{y}}{\lambda}\right), \ \ \forall\, \lambda > 0. \tag{47}$$

Compared to (6), the only difference in (47) is that the feasible set $\overline{F}$ is the intersection of a set of ellipsoids, and thus not a polytope. However, similar to F, $\overline{F}$ is also a nonempty closed and convex (notice that 0 is a feasible point). Therefore, we can make use of all the aforementioned properties of the projection operators, e.g., Lemmas 9 and 13, Theorems 12 and 15, to develop screening rules for the group Lasso problem.

Moreover, similar to the case of Lasso, we also have a specific parameter value (Tibshirani et al., 2012) for the group Lasso problem, i.e.,

$$\overline{\lambda}_{\max} = \max_g \frac{\|\mathbf{X}_g^T\mathbf{y}\|_2}{\sqrt{n_g}}. \tag{48}$$

Indeed, $\overline{\lambda}_{\max}$ is the smallest parameter value such that the optimal solution of problem (43) is 0. More specifically, we have:

$$\beta^*(\lambda) = 0, \ \ \forall\, \lambda \in [\overline{\lambda}_{\max}, \infty). \tag{49}$$

Combining the result in (49) and (45), we immediately have

$$\theta^*(\lambda) = \frac{\mathbf{y}}{\lambda}, \ \ \forall\, \lambda \in [\overline{\lambda}_{\max}, \infty). \tag{50}$$

Therefore, all through the subsequent sections, we will focus on the cases with $\lambda \in (0, \overline{\lambda}_{\max})$.

3.2 Enhanced DPP rule for Group Lasso

In view of (R2′), we can see that the estimation of the dual optimal solution is the key step to develop a screening rule for the group Lasso problem. Because $\theta^*(\lambda)$ is the projection of $\mathbf{y}/\lambda$ onto the nonempty closed convex set $\overline{F}$ [please refer to (47)], we can make use of all the properties of projection operators, e.g., Lemmas 9 and 13, Theorems 12 and 15, to estimate the dual optimal solution. First, let us develop a useful technical result as follows.

Lemma 21 *For the group Lasso problem, let $\overline{\lambda}_{\max}$ be given by (48) and*

$$\mathbf{X}_* := \mathrm{argmax}_{\mathbf{X}_g} \frac{\|\mathbf{X}_g^T\mathbf{y}\|_2}{\sqrt{n_g}}. \tag{51}$$

Suppose that the dual optimal solution $\theta^*(\cdot)$ is known at $\lambda_0 \in (0, \overline{\lambda}_{\max}]$, let us define

$$\begin{aligned} \overline{\mathbf{v}}_1(\lambda_0) &= \begin{cases} \dfrac{\mathbf{y}}{\lambda_0} - \theta^*(\lambda_0), & \text{if } \lambda_0 \in (0, \overline{\lambda}_{\max}), \\ \mathbf{X}_* \mathbf{X}_*^T \mathbf{y}, & \text{if } \lambda_0 = \overline{\lambda}_{\max}. \end{cases} \\ \overline{\theta}_{\lambda_0}(t) &= \theta^*(\lambda_0) + t\overline{\mathbf{v}}_1(\lambda_0), \quad t \geq 0. \end{aligned} \tag{52}$$

Then, we have the following result holds

$$P_{\overline{F}}(\overline{\theta}_{\lambda_0}(t)) = \theta^*(\lambda_0), \ \forall\ t \geq 0. \tag{53}$$

Proof Let us first consider the cases with $\lambda_0 \in (0, \overline{\lambda}_{\max})$. In view of the definition of $\overline{\lambda}_{\max}$, it is easy to see that $\mathbf{y}/\lambda_0 \notin \overline{F}$. Therefore, in view of (47) and Lemma 9, the statement in (53) follows immediately.

We next consider the case with $\lambda_0 = \overline{\lambda}_{\max}$. By Theorem 12, we only need to check if

$$\overline{\mathbf{v}}_1(\overline{\lambda}_{\max}) \in N_{\overline{F}}(\theta^*(\overline{\lambda}_{\max})) \Leftrightarrow \left\langle \overline{\mathbf{v}}_1(\overline{\lambda}_{\max}), \theta - \theta^*(\overline{\lambda}_{\max}) \right\rangle \leq 0, \ \ \forall\, \theta \in \overline{F}. \tag{54}$$

Indeed, in view of (48) and (50), we can see that

$$\langle \overline{\mathbf{v}}_1(\overline{\lambda}_{\max}), \theta^*(\overline{\lambda}_{\max}) \rangle = \left\langle \mathbf{X}_* \mathbf{X}_*^T \mathbf{y}, \frac{\mathbf{y}}{\overline{\lambda}_{\max}} \right\rangle = \frac{\|\mathbf{X}_*^T \mathbf{y}\|_2^2}{\overline{\lambda}_{\max}}. \tag{55}$$

On the other hand, by (48) and (51), we can see that

$$\|\mathbf{X}_*^T \mathbf{y}\|_2 = \overline{\lambda}_{\max} \sqrt{n_*}, \tag{56}$$

where n_* is the number of columns of $\mathbf{X}_*$. By plugging (56) into (55), we have

$$\langle \overline{\mathbf{v}}_1(\overline{\lambda}_{\max}), \theta^*(\overline{\lambda}_{\max}) \rangle = \overline{\lambda}_{\max} \cdot n_*.$$

Moreover, for any feasible point $\theta \in \overline{F}$, we can see that

$$\|\mathbf{X}_*^T \theta\|_2 \leq \sqrt{n_*}. \tag{57}$$

In view of the result in (57) and (56), it is easy to see that

$$\left\langle \overline{\mathbf{v}}_1(\overline{\lambda}_{\max}), \theta \right\rangle = \left\langle \mathbf{X}_* \mathbf{X}_*^T \mathbf{y}, \theta \right\rangle = \left\langle \mathbf{X}_*^T \mathbf{y}, \mathbf{X}_*^T \theta \right\rangle \leq \|\mathbf{X}_*^T \mathbf{y}\|_2 \|\mathbf{X}_*^T \theta\|_2 = \overline{\lambda}_{\max} \cdot n_*. \tag{58}$$

Combining the result in (55) and (58), it is easy to see that the inequality in (54) holds for all $\theta \in \overline{F}$, which completes the proof. ■

By Lemma 21, we can accurately estimate the dual optimal solution of the group Lasso problem in the following theorem. It is easy to see that the result in Theorem 22 is very similar to the one in Theorem 18 for the Lasso problem.

Theorem 22 *For the group Lasso problem, suppose that the dual optimal solution $\theta^*(\cdot)$ at $\theta_0 \in (0, \overline{\lambda}_{\max}]$ is known, and $\overline{\mathbf{v}}_1(\lambda_0)$ is given by (52). For any $\lambda \in (0, \lambda_0]$, let us define*

$$\overline{\mathbf{v}}_2(\lambda, \lambda_0) = \frac{\mathbf{y}}{\lambda} - \theta^*(\lambda_0),$$

$$\overline{\mathbf{v}}_2^{\perp}(\lambda, \lambda_0) = \overline{\mathbf{v}}_2(\lambda, \lambda_0) - \frac{\langle \overline{\mathbf{v}}_1(\lambda_0), \overline{\mathbf{v}}_2(\lambda, \lambda_0)\rangle}{\|\overline{\mathbf{v}}_1(\lambda_0)\|_2^2}\overline{\mathbf{v}}_1(\lambda_0).$$

Then, the dual optimal solution $\theta^*(\lambda)$ can be estimated as follows:

$$\left\|\theta^*(\lambda) - \left(\theta^*(\lambda_0) + \frac{1}{2}\overline{\mathbf{v}}_2^{\perp}(\lambda, \lambda_0)\right)\right\|_2 \leq \frac{1}{2}\|\overline{\mathbf{v}}_2^{\perp}(\lambda, \lambda_0)\|_2.$$

We omit the proof of Theorem 22 since it is exactly the same as the one of Theorem 18. Indeed, Theorem 22 is equivalent to estimating $\theta^*(\lambda)$ in a ball as follows:

$$\theta^*(\lambda) \in B\left(\theta^*(\lambda_0) + \frac{1}{2}\overline{\mathbf{v}}_2^{\perp}(\lambda, \lambda_0), \frac{1}{2}\|\overline{\mathbf{v}}_2^{\perp}(\lambda, \lambda_0)\|_2\right). \tag{59}$$

Based on this estimation and (R2′), we immediately have the following result.

Theorem 23 *For the group Lasso problem, assume the dual optimal solution $\theta^*(\cdot)$ is known at $\lambda_0 \in (0, \overline{\lambda}_{\max}]$, and $\lambda \in (0, \lambda_0]$. Then $\beta_g^*(\lambda) = 0$ if the following holds*

$$\left\|\mathbf{X}_g^T\left(\theta^*(\lambda_0) + \frac{1}{2}\overline{\mathbf{v}}_2^{\perp}(\lambda, \lambda_0)\right)\right\|_2 < \sqrt{n_g} - \frac{1}{2}\|\overline{\mathbf{v}}_2^{\perp}(\lambda, \lambda_0)\|_2\|\mathbf{X}_g\|_2. \tag{60}$$

Proof In view of (R2′), we only need to check if

$$\left\|\mathbf{X}_g^T\theta^*(\lambda)\right\|_2 < \sqrt{n_g}.$$

To simplify notations, let

$$\mathbf{o} = \theta^*(\lambda_0) + \frac{1}{2}\overline{\mathbf{v}}_2^{\perp}(\lambda, \lambda_0), \;\; r = \frac{1}{2}\|\overline{\mathbf{v}}_2^{\perp}(\lambda, \lambda_0)\|_2.$$

It is easy to see that

$$\begin{aligned}\left\|\mathbf{X}_g^T\theta^*(\lambda)\right\|_2 &\leq \|\mathbf{X}_g^T(\theta^*(\lambda) - \mathbf{o})\|_2 + \|\mathbf{X}_g^T\mathbf{o}\|_2 \\ &< \|\mathbf{X}_g\|_2\|\theta^*(\lambda) - \mathbf{o}\|_2 + \sqrt{n_g} - r\|\mathbf{X}_g\|_2 \\ &\leq r\|\mathbf{X}_g\|_2 + \sqrt{n_g} - r\|\mathbf{X}_g\|_2 = \sqrt{n_g},\end{aligned} \tag{61}$$

which completes the proof. The second and third inequalities in (61) are due to (60) and Theorem 22, respectively. ■

In view of (45) and Theorem 23, we can derive the EDPP rule to discard the inactive groups for the group Lasso problem as follows.

Corollary 24 EDPP: *For the group Lasso problem (43), suppose we are given a sequence of parameter values $\overline{\lambda}_{\max} = \lambda_0 > \lambda_1 > \ldots > \lambda_{\mathcal{K}}$. For any integer $0 \leq k < \mathcal{K}$, we have $\beta_g^*(\lambda_{k+1}) = 0$ if $\beta^*(\lambda_k)$ is known and the following holds:*

$$\left\|\mathbf{X}_g^T\left(\frac{\mathbf{y} - \sum_{g=1}^G \mathbf{X}_g\beta_g^*(\lambda_k)}{\lambda_k} + \frac{1}{2}\overline{\mathbf{v}}_2^{\perp}(\lambda_{k+1}, \lambda_k)\right)\right\|_2 < \sqrt{n_g} - \frac{1}{2}\|\overline{\mathbf{v}}_2^{\perp}(\lambda_{k+1}, \lambda_k)\|_2\|\mathbf{X}_g\|_2.$$

4. Experiments

In this section, we evaluate the proposed EDPP rules for Lasso and group Lasso on both synthetic and real data sets. To measure the performance of our screening rules, we compute the *rejection ratio* and *speedup* (please refer to Section 2.3.3 for details). Because the EDPP rule is safe, i.e., no active features/groups will be mistakenly discarded, the rejection ratio will be less than one.

In Section 4.1, we conduct two sets of experiments to compare the performance of EDPP against several state-of-the-art screening methods. We first compare the performance of the basic versions of EDPP, DOME, SAFE, and strong rule. Then, we focus on the sequential versions of EDPP, SAFE, and strong rule. Notice that, SAFE and EDPP are safe. However, strong rule may mistakenly discard features with nonzero coefficients in the solution. Although DOME is also safe for the Lasso problem, it is unclear if there exists a sequential version of DOME. Recall that, real applications usually favor the sequential screening rules because we need to solve a sequence of of Lasso problems to determine an appropriate parameter value (Tibshirani et al., 2012). Moreover, DOME assumes special structure on the data, i.e., each feature and the response vector should be normalized to have unit length.

In Section 4.2, we compare EDPP with strong rule for the group Lasso problem on synthetic data sets. We are not aware of any *safe* screening rules for the group Lasso problem at this point. For SAFE and Dome, it is not straightforward to extend them to the group Lasso problem.

An efficient MATLAB implementation of the EDPP screening rules—combined with the solvers from SLEP package (Liu et al., 2009)—for both Lasso and group Lasso is available at `http://dpc-screening.github.io/`.

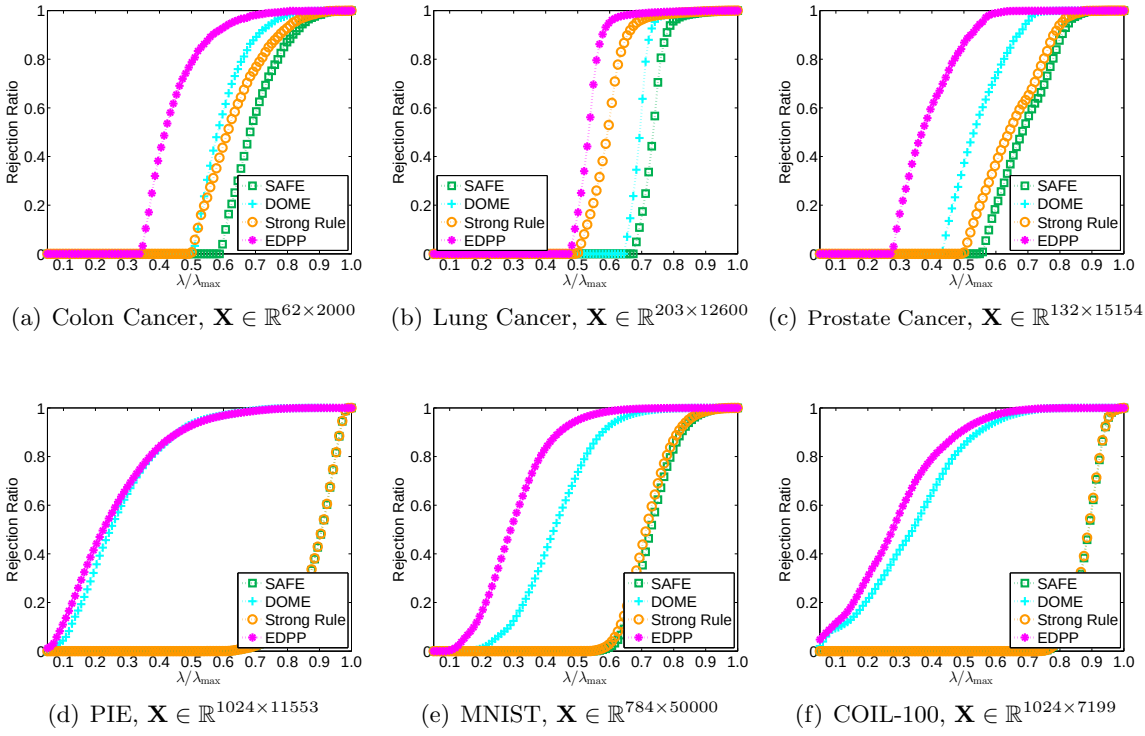
4.1 EDPP for the Lasso Problem

For the Lasso problem, we first compare the performance of the basic versions of EDPP, DOME, SAFE and strong rule in Section 4.1.1. Then, we compare the performance of the sequential versions of EDPP, SAFE and strong rule in Section 4.1.2.

4.1.1 Evaluation of the Basic EDPP Rule

In this section, we perform experiments on six real data sets to compare the performance of the basic versions of SAFE, DOME, strong rule and EDPP. Briefly speaking, suppose that we are given a parameter value λ. Basic versions of the aforementioned screening rules always make use of $\beta^*(\lambda_{\max})$ to identify the zero components of $\beta^*(\lambda)$. Take EDPP for example. The basic version of EDPP can be obtained by replacing $\beta^*(\lambda_k)$ and $\mathbf{v}_2^\perp(\lambda_{k+1}, \lambda_k)$ with $\beta^*(\lambda_0)$ and $\mathbf{v}_2^\perp(\lambda_k, \lambda_0)$, respectively, in (42) for all $k = 1, \ldots, \mathcal{K}$.

In this experiment, we report the rejection ratios of the basic SAFE, DOME, strong rule and EDPP along a sequence of 100 parameter values equally spaced on the $\lambda/\lambda_{\max}$ scale from 0.05 to 1.0. We note that DOME requires that all features of the data sets have unit length. Therefore, to compare the performance of DOME with SAFE, strong rule and EDPP, we normalize the features of all the data sets used in this section. However,



(a) Colon Cancer, $\mathbf{X} \in \mathbb{R}^{62\times 2000}$ (b) Lung Cancer, $\mathbf{X} \in \mathbb{R}^{203\times 12600}$ (c) Prostate Cancer, $\mathbf{X} \in \mathbb{R}^{132\times 15154}$

(d) PIE, $\mathbf{X} \in \mathbb{R}^{1024\times 11553}$ (e) MNIST, $\mathbf{X} \in \mathbb{R}^{784\times 50000}$ (f) COIL-100, $\mathbf{X} \in \mathbb{R}^{1024\times 7199}$

Figure 2: Comparison of basic versions of SAFE, DOME, Strong Rule and EDPP on six real data sets.

it is worthwhile to mention that SAFE, strong rule and EDPP do not assume any specific structures on the data set. The data sets used in this section are listed as follows:

a) Colon Cancer data set (Alon et al., 1999);

b) Lung Cancer data set (Bhattacharjee et al., 2001);

c) Prostate Cancer data set (Petricoin et al., 2002);

d) PIE face image data set (Sim et al., 2003; Cai et al., 2007);

e) MNIST handwritten digit data set (Lecun et al., 1998);

f) COIL-100 image data set (Nene et al., 1996; Cai et al., 2011).

The Colon Cancer data set contains gene expression information of 22 normal tissues and 40 colon cancer tissues, and each has 2000 gene expression values.

The Lung Cancer data set contains gene expression information of 186 lung tumors and 17 normal lung specimens. Each specimen has 12600 expression values.

The COIL-100 image data set consists of images of 100 objects. The images of each object are taken every 5 degree by rotating the object, yielding 72 images per object. The

dimension of each image is 32×32. In each trial, we randomly select one image as the response vector and use the remaining ones as the data matrix. We run 100 trials and report the average performance of the screening rules.

The description and the experimental settings for the Prostate Cancer data set, the PIE face image data set and the MNIST handwritten digit data set are given in Section 2.3.3.

Figure 2 reports the rejection ratios of the basic versions of SAFE, DOME, strong rule and EDPP. We can see that EDPP significantly outperforms the other three screening methods on five of the six data sets, i.e., the Colon Cancer, Lung Cancer, Prostate Cancer, MNIST, and COIL-100 data sets. On the PIE face image data set, EDPP and DOME provide similar performance and both significantly outperform SAFE and strong rule.

However, as pointed out by Tibshirani et al. (2012), the real strength of screening methods stems from their sequential versions. The reason is because the optimal parameter value is unknown in real applications. Typical approaches for model selection usually involve solving the Lasso problems many times along a sequence of parameter values. Thus, the sequential screening methods are more suitable in facilitating the aforementioned scenario and more useful than their basic-version counterparts in practice (Tibshirani et al., 2012).

4.1.2 Evaluation of the Sequential EDPP Rule

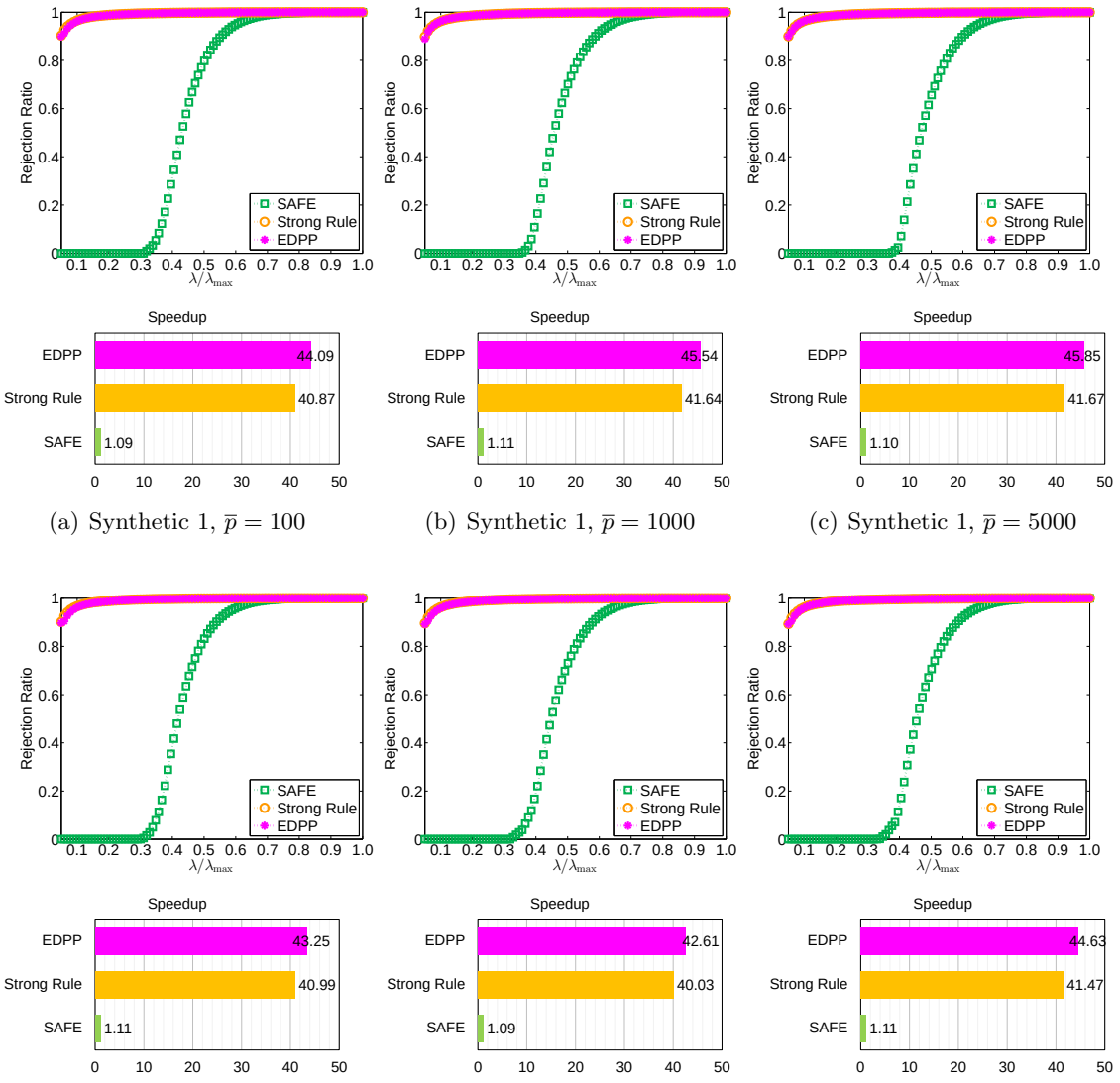
In this section, we compare the performance of the sequential versions of SAFE, strong rule and EDPP by the rejection ratio and speedup. We first perform experiments on two synthetic data sets. We then apply the three screening rules to six real data sets.

We first perform experiments on several synthetic problems, which have been commonly used in the sparse learning literature (Bondell and Reich, 2008; Zou and Hastie, 2005; Tibshirani, 1996). We simulate data from the true model

$$\mathbf{y} = \mathbf{X}\beta^* + \sigma\epsilon, \;\; \epsilon \sim N(0,1). \tag{62}$$

We generate two data sets with 250×10000 entries: Synthetic 1 and Synthetic 2. For Synthetic 1, the entries of the data matrix $\mathbf{X}$ are i.i.d. standard Gaussian with pairwise correlation zero, i.e., $\mathrm{corr}(\mathbf{x}_i, \mathbf{x}_i) = 0$. For Synthetic 2, the entries of the data matrix $\mathbf{X}$ are drawn from i.i.d. standard Gaussian with pairwise correlation $0.5^{|i-j|}$, i.e., $\mathrm{corr}(\mathbf{x}_i, \mathbf{x}_j) = 0.5^{|i-j|}$. To generate the response vector $\mathbf{y} \in \mathbb{R}^{250}$ by the model in (62), we need to set the parameter σ and construct the ground truth $\beta^* \in \mathbb{R}^{10000}$. Throughout this section, σ is set to be 0.1. To construct β^*, we randomly select $\overline{p}$ components which are populated from a uniform $[-1, 1]$ distribution, and set the remaining ones as 0. After we generate the data matrix $\mathbf{X}$ and the response vector $\mathbf{y}$, we run the solver with or without screening rules to solve the Lasso problems along a sequence of 100 parameter values equally spaced on the $\lambda/\lambda_{\max}$ scale from 0.05 to 1.0. We then run 100 trials and report the average performance.

We first apply the screening rules, i.e., SAFE, strong rule and EDPP to Synthetic 1 with $\overline{p} = 100, 1000, 5000$ respectively. Figure 3(a), Figure 3(b) and Figure 3(c) present the corresponding rejection ratios and speedup of SAFE, strong rule and EDPP. We can see that the rejection ratios of strong rule and EDPP are comparable to each other, and both of them are more effective in discarding inactive features than SAFE. In terms of the speedup, EDPP provides better performance than strong rule. The reason is because strong rule is a heuristic screening method, i.e., it may mistakenly discard active features which have



(a) Synthetic 1, $\overline{p} = 100$ (b) Synthetic 1, $\overline{p} = 1000$ (c) Synthetic 1, $\overline{p} = 5000$

(d) Synthetic 2, $\overline{p} = 100$ (e) Synthetic 2, $\overline{p} = 1000$ (f) Synthetic 2, $\overline{p} = 5000$

Figure 3: Comparison of SAFE, Strong Rule and EDPP on two synthetic data sets with different numbers of nonzero components of the ground truth.

nonzero components in the solution. Thus, strong rule needs to check the KKT conditions to ensure the correctness of the screening result. In contrast, the EDPP rule does not need to check the KKT conditions since the discarded features are guaranteed to be absent from the resulting sparse representation. From the last two columns of Table 2, we can observe that the running time of strong rule is about twice of that of EDPP.

Figure 3(d), Figure 3(e) and Figure 3(f) present the rejection ratios and speedup of SAFE, strong rule and EDPP on Synthetic 2 with $\overline{p} = 100, 1000, 5000$ respectively. We can

Data	$\overline{p}$	solver	SAFE+solver	Strong Rule+solver	EDPP+solver	SAFE	Strong Rule	EDPP
Synthetic 1	100	109.01	100.09	2.67	2.47	4.60	0.65	0.36
	1000	123.60	111.32	2.97	2.71	4.59	0.66	0.37
	5000	124.92	113.09	3.00	2.72	4.57	0.65	0.36
Synthetic 2	100	107.50	96.94	2.62	2.49	4.61	0.67	0.37
	1000	113.59	104.29	2.84	2.67	4.57	0.63	0.35
	5000	125.25	113.35	3.02	2.81	4.62	0.65	0.36

Table 2: Running time (in seconds) for solving the Lasso problems along a sequence of 100 tuning parameter values equally spaced on the scale of $\lambda/\lambda_{\max}$ from 0.05 to 1 by (a): the solver (Liu et al., 2009) (reported in the third column) without screening; (b): the solver combined with different screening methods (reported in the 4^{th} to the 6^{th} columns). The last four columns report the total running time (in seconds) for the screening methods.

observe patterns similar to Synthetic 1. Clearly, our method, EDPP, is very robust to the variations of the intrinsic structures of the data sets and the sparsity of the ground truth.

We next compare the performance of the EDPP rule with SAFE and strong rule on six real data sets along a sequence of 100 parameter values equally spaced on the $\lambda/\lambda_{\max}$ scale from 0.05 to 1.0. The data sets are listed as follows:

a) Breast Cancer data set (West et al., 2001; Shevade and Keerthi, 2003);

b) Leukemia data set (Armstrong et al., 2002);

c) Prostate Cancer data set (Petricoin et al., 2002);

d) PIE face image data set (Sim et al., 2003; Cai et al., 2007);

e) MNIST handwritten digit data set (Lecun et al., 1998);

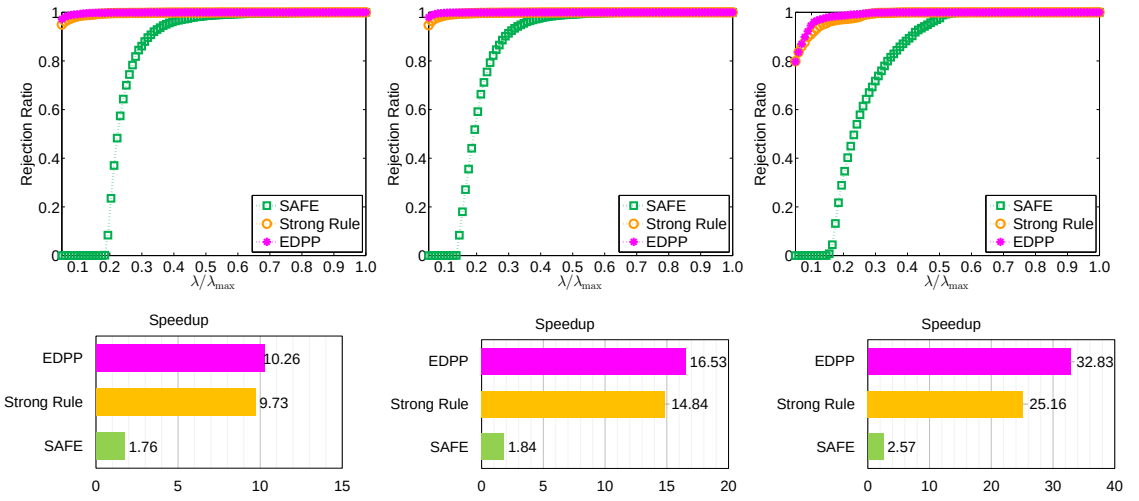
f) Street View House Number (SVHN) data set (Netzer et al., 2001).

We present the rejection ratios and speedup of EDPP, SAFE and strong rule in Figure 4. Table 3 reports the running time of the solver with or without screening for solving the 100 Lasso problems, and that of the screening rules.

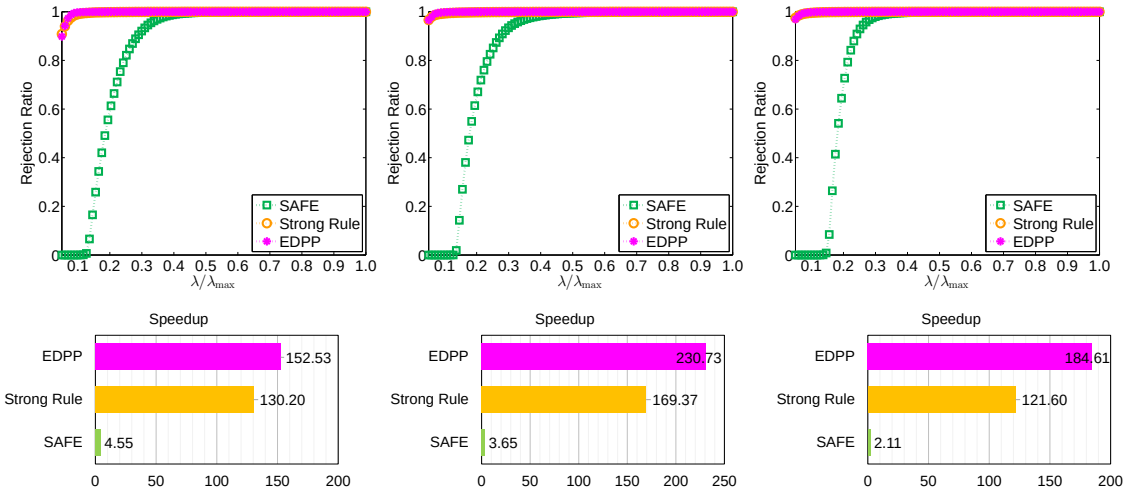
The Breast Cancer data set contains 44 tumor samples, each of which is represented by 7129 genes. Therefore, the data matrix $\mathbf{X}$ is of 44×7129. The response vector $\mathbf{y} \in \{1, -1\}^{44}$ contains the binary label of each sample.

The Leukemia data set is a DNA microarray data set, containing 52 samples and 11225 genes. Therefore, the data matrix $\mathbf{X}$ is of 52×11225. The response vector $\mathbf{y} \in \{1, -1\}^{52}$ contains the binary label of each sample.

The SVHN data set contains color images of street view house numbers, including 73257 images for training and 26032 for testing. The dimension of each image is 32×32.



(a) Breast Cancer, $\mathbf{X} \in \mathbb{R}^{44\times 7129}$ (b) Leukemia, $\mathbf{X} \in \mathbb{R}^{52\times 11225}$ (c) Prostate Cancer, $\mathbf{X} \in \mathbb{R}^{132\times 15154}$



(d) PIE, $\mathbf{X} \in \mathbb{R}^{1024\times 11553}$ (e) MNIST, $\mathbf{X} \in \mathbb{R}^{784\times 50000}$ (f) SVHN, $\mathbf{X} \in \mathbb{R}^{3072\times 99288}$

Figure 4: Comparison of SAFE, Strong Rule, and EDPP on six real data sets.

In each trial, we first randomly select an image as the response $\mathbf{y} \in \mathbb{R}^{3072}$, and then use the remaining ones to form the data matrix $\mathbf{X} \in \mathbb{R}^{3072\times 99288}$. We run 100 trials and report the average performance.

The description and the experiment settings for the Prostate Cancer data set, the PIE face image data set and the MNIST handwritten digit data set are given in Section 2.3.3.

From Figure 4, we can see that the rejection ratios of strong rule and EDPP are comparable to each other. Compared to SAFE, both of strong rule and EDPP are able to identify far more inactive features, leading to a much higher speedup. However, because strong rule needs to check the KKT conditions to ensure the correctness of the screening

Data	solver	SAFE+solver	Strong Rule+solver	EDPP+solver	SAFE	Strong Rule	EDPP
Breast Cancer	12.70	7.20	1.31	1.24	0.44	0.06	0.05
Leukemia	16.99	9.22	1.15	1.03	0.91	0.09	0.07
Prostate Cancer	121.41	47.17	4.83	3.70	3.60	0.46	0.23
PIE	629.94	138.33	4.84	4.13	19.93	2.54	1.33
MNIST	2566.26	702.21	15.15	11.12	64.81	8.14	4.19
SVHN	11023.30	5220.88	90.65	59.71	583.12	61.02	31.64

Table 3: Running time (in seconds) for solving the Lasso problems along a sequence of 100 tuning parameter values equally spaced on the scale of $\lambda/\lambda_{\max}$ from 0.05 to 1 by (a): the solver (Liu et al., 2009) (reported in the second column) without screening; (b): the solver combined with different screening methods (reported in the 3^{rd} to the 5^{th} columns). The last three columns report the total running time (in seconds) for the screening methods.

results, the speedup gained by EDPP is higher than that by strong rule. When the size of the data matrix is not very large, e.g., the Breast Cancer and Leukemia data sets, the speedup gained by EDPP are slightly higher than that by strong rule. However, when the size of the data matrix is large, e.g., the MNIST and SVHN data sets, the speedup gained by EDPP are significantly higher than that by strong rule. Moreover, we can also observe from Figure 4 that, the larger the data matrix is, the higher the speedup can be gained by EDPP. More specifically, for the small data sets, e.g., the Breast Cancer, Leukemia and Prostate Cancer data sets, the speedup gained by EDPP is about 10, 17 and 30 times. In contrast, for the large data sets, e.g., the PIE, MNIST and SVHN data sets, the speedup gained by EDPP is two orders of magnitude. Take the SVHN data set for example. The solver without screening needs about 3 *hours* to solve the 100 Lasso problems. Combined with the EDPP rule, the solver only needs less than 1 *minute* to complete the task.

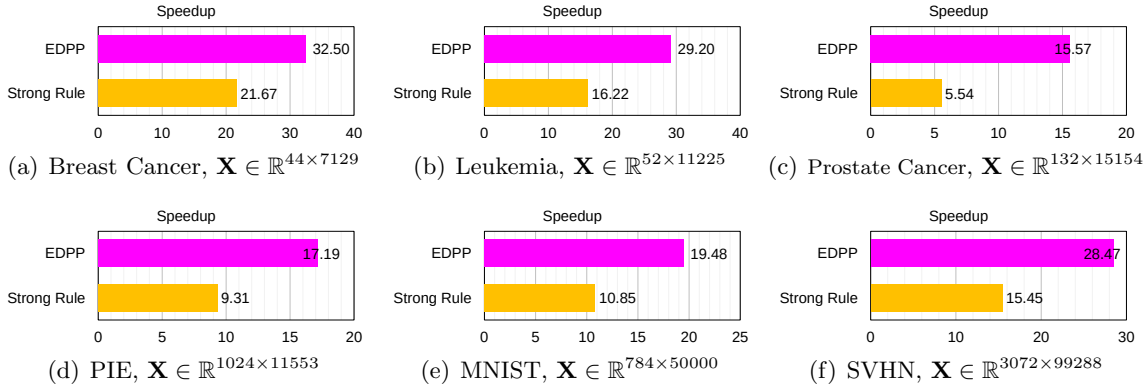
Clearly, the proposed EDPP screening rule is very effective in accelerating the computation of Lasso especially for large-scale problems, and outperforms the state-of-the-art approaches like SAFE and strong rule. Notice that, the EDPP method is safe in the sense that the discarded features are guaranteed to have zero coefficients in the solution.

4.1.3 EDPP with Least-Angle Regression (LARS)

As we mentioned in the introduction, we can combine EDPP with any existing solver. In this experiment, we integrate EDPP and strong rule with another state-of-the-art solver for Lasso, i.e., Least-Angle Regression (LARS) (Efron et al., 2004). We perform experiments on the same real data sets used in the last section with the same experiment settings. Because the rejection ratios of screening methods are irrelevant to the solvers, we only report the speedup. Table 4 reports the running time of LARS with or without screening for solving the 100 Lasso problems, and that of the screening methods. Figure 5 shows the speedup of these two methods. We can still observe a substantial speedup gained by EDPP. The

Data	LARS	Strong Rule+LARS	EDPP+LARS	Strong Rule	EDPP
Breast Cancer	1.30	0.06	0.04	0.04	0.03
Leukemia	1.46	0.09	0.05	0.07	0.04
Prostate Cancer	5.76	1.04	0.37	0.42	0.24
PIE	22.52	2.42	1.31	2.30	1.21
MNIST	92.53	8.53	4.75	8.36	4.34
SVHN	1017.20	65.83	35.73	62.53	32.00

Table 4: Running time (in seconds) for solving the Lasso problems along a sequence of 100 tuning parameter values equally spaced on the scale of $\lambda/\lambda_{\max}$ from 0.05 to 1 by (a): the solver (Efron et al., 2004; Mairal et al., 2010) (reported in the second column) without screening; (b): the solver combined with different screening methods (reported in the 3^{rd} and 4^{th} columns). The last two columns report the total running time (in seconds) for the screening methods.



(a) Breast Cancer, $\mathbf{X} \in \mathbb{R}^{44\times 7129}$ (b) Leukemia, $\mathbf{X} \in \mathbb{R}^{52\times 11225}$ (c) Prostate Cancer, $\mathbf{X} \in \mathbb{R}^{132\times 15154}$

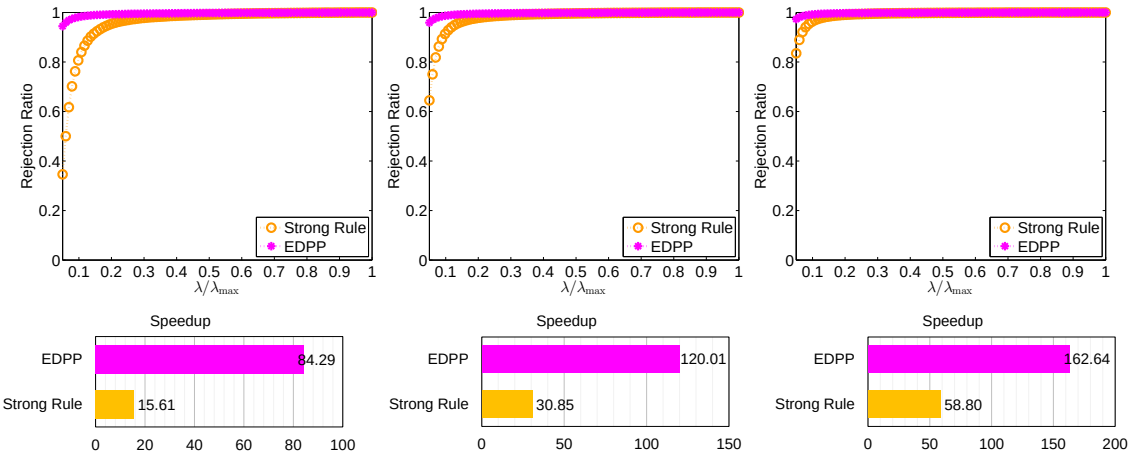
(d) PIE, $\mathbf{X} \in \mathbb{R}^{1024\times 11553}$ (e) MNIST, $\mathbf{X} \in \mathbb{R}^{784\times 50000}$ (f) SVHN, $\mathbf{X} \in \mathbb{R}^{3072\times 99288}$

Figure 5: The speedup gained by Strong Rule and EDPP combined with LARS on six real data sets.

reason is that EDPP has a very low computational cost (see Table 4) and it is very effective in discarding inactive features (see Figure 4).

4.2 EDPP for the Group Lasso Problem

In this experiment, we evaluate the performance of EDPP and strong rule with different numbers of groups. The data matrix $\mathbf{X}$ is fixed to be 250×200000. The entries of the response vector $\mathbf{y}$ and the data matrix $\mathbf{X}$ are generated i.i.d. from a standard Gaussian distribution. For each experiment, we repeat the computation 20 times and report the average results. Moreover, let n_g denote the number of groups and s_g be the average group size. For example, if n_g is 10000, then $s_g = p/n_g = 20$.



(a) $n_g = 10000$ (b) $n_g = 20000$ (c) $n_g = 40000$

Figure 6: Comparison of EDPP and strong rules with different numbers of groups.

n_g	solver	Strong Rule+solver	EDPP+solver	Strong Rule	EDPP
10000	4535.54	296.60	53.81	13.99	8.32
20000	5536.18	179.48	46.13	14.16	8.61
40000	6144.48	104.50	37.78	13.13	8.37

Table 5: Running time (in seconds) for solving the group Lasso problems along a sequence of 100 tuning parameter values equally spaced on the scale of $\lambda/\lambda_{\max}$ from 0.05 to 1.0 by (a): the solver from SLEP (reported in the second column) without screening; (b): the solver combined with different screening methods (reported in the 3^{rd} and 4^{th} columns). The last two columns report the total running time (in seconds) for the screening methods. The data matrix $\mathbf{X}$ is of size 250×200000.

From Figure 6, we can see that EDPP and strong rule are able to discard more inactive groups when the number of groups n_g increases. The intuition behind this observation is that the estimation of the dual optimal solution is more accurate with a smaller group size. Notice that, a large n_g implies a small average group size. Figure 6 also implies that compared to strong rule, EDPP is able to discard more inactive groups and is more robust with respect to different values of n_g.

Table 5 further demonstrates the effectiveness of EDPP in improving the efficiency of the solver. When $n_g = 10000$, the efficiency of the solver is improved by about 80 times. When $n_g = 20000$ and 40000, the efficiency of the solver is boosted by about 120 and 160 times with EDPP respectively.

5. Conclusion

In this paper, we develop new screening rules for the Lasso problem by making use of the properties of the projection operators with respect to a closed convex set. Our proposed methods, i.e., DPP screening rules, are able to effectively identify inactive predictors of the Lasso problem, thus greatly reducing the size of the optimization problem. Moreover, we further improve DPP rule and propose the enhanced DPP rule, which is more effective in discarding inactive features than DPP rule. The idea of the family of DPP rules can be easily generalized to identify the inactive groups of the group Lasso problem. Extensive numerical experiments on both synthetic and real data demonstrate the effectiveness of the proposed rules. It is worthwhile to mention that the family of DPP rules can be combined with any Lasso solver as a speedup tool. In the future, we plan to generalize our ideas to other sparse formulations consisting of more general structured sparse penalties, e.g., tree/graph Lasso, fused Lasso.

Acknowledgments

We would like to acknowledge support for this project from the National Science Foundation (IIS-0953662, IIS-1421057, and IIS-1421100) and the National Institutes of Health (R01 LM010730 and U54 EB020403).

Appendix A. The Dual Problem of Lasso

In this appendix, we give the detailed derivation of the dual problem of Lasso.

A.1 Dual Formulation

Assuming the data matrix is $\mathbf{X} \in \mathbb{R}^{N\times p}$, the standard Lasso problem is given by:

$$\inf_{\beta\in\mathbb{R}^p} \frac{1}{2}\|\mathbf{y}-\mathbf{X}\beta\|_2^2 + \lambda\|\beta\|_1. \tag{63}$$

For completeness, we give a detailed deviation of the dual formulation of (63) in this section. Note that problem (63) has no constraints. Therefore the dual problem is trivial and useless. A common trick (Boyd and Vandenberghe, 2004) is to introduce a new set of variables $\mathbf{z} = \mathbf{y} - \mathbf{X}\beta$ such that problem (63) becomes:

$$\begin{aligned} \inf_{\beta} \quad & \frac{1}{2}\|\mathbf{z}\|_2^2 + \lambda\|\beta\|_1, \\ \text{subject to} \quad & \mathbf{z} = \mathbf{y} - \mathbf{X}\beta. \end{aligned} \tag{64}$$

By introducing the dual variables $\eta \in \mathbb{R}^N$, we get the Lagrangian of problem (64):

$$L(\beta, \mathbf{z}, \eta) = \frac{1}{2}\|\mathbf{z}\|_2^2 + \lambda\|\beta\|_1 + \eta^T \cdot (\mathbf{y} - \mathbf{X}\beta - \mathbf{z}).$$

For the Lagrangian, the primal variables are β and $\mathbf{z}$. And the dual function $g(\eta)$ is:

$$g(\eta) = \inf_{\beta,\mathbf{z}} L(\beta, \mathbf{z}, \eta) = \eta^T\mathbf{y} + \inf_{\beta}(-\eta^T\mathbf{X}\beta + \lambda\|\beta\|_1) + \inf_{\mathbf{z}} \big(\frac{1}{2}\|\mathbf{z}\|_2^2 - \eta^T\mathbf{z}\big).$$

In order to get $g(\eta)$, we need to solve the following two optimization problems.

$$\inf_{\beta} -\eta^T \mathbf{X}\beta + \lambda\|\beta\|_1, \tag{65}$$

and

$$\inf_{\mathbf{z}} \frac{1}{2}\|\mathbf{z}\|_2^2 - \eta^T \mathbf{z}. \tag{66}$$

Let us first consider problem (65). Denote the objective function of problem (65) as

$$f_1(\beta) = -\eta^T \mathbf{X}\beta + \lambda\|\beta\|_1. \tag{67}$$

$f_1(\beta)$ is convex but not smooth. Therefore let us consider its subgradient

$$\partial f_1(\beta) = -\mathbf{X}^T\eta + \lambda\mathbf{v},$$

in which $\|\mathbf{v}\|_\infty \leq 1$ and $\mathbf{v^T}\beta = \|\beta\|_1$, i.e., $\mathbf{v}$ is the subgradient of $\|\beta\|_1$.

The necessary condition for f_1 to attain an optimum is

$$\exists\, \beta', \text{ such that } 0 \in \partial f_1(\beta') = \{-\mathbf{X}^T\eta + \lambda\mathbf{v}'\},$$

where $\mathbf{v}' \in \partial\|\beta'\|_1$. In other words, $\beta', \mathbf{v}'$ should satisfy

$$\mathbf{v}' = \frac{\mathbf{X}^T\eta}{\lambda}, \|\mathbf{v}'\|_\infty \leq 1, \mathbf{v'}^T\beta' = \|\beta'\|_1,$$

which is equivalent to

$$|\mathbf{x}_i^T\eta| \leq \lambda, i = 1, 2, \ldots, p.$$

Then we plug $\mathbf{v}' = \frac{\mathbf{X}^T\eta}{\lambda}$ and $\mathbf{v'}^T\beta' = \|\beta'\|_1$ into (67):

$$f_1(\beta') = \inf_{\beta} f_1(\beta) = -\eta^T\mathbf{X}\beta' + \lambda\big(\frac{\mathbf{X}^T\eta}{\lambda}\big)^T\beta' = 0.$$

Therefore, the optimum value of problem (65) is 0.

Next, let us consider problem (66). Denote the objective function of problem (66) as $f_2(\mathbf{z})$. Let us rewrite $f_2(\mathbf{z})$ as:

$$f_2(\mathbf{z}) = \frac{1}{2}(\|\mathbf{z} - \eta\|_2^2 - \|\eta\|_2^2).$$

Clearly,

$$\mathbf{z}' = \operatorname*{argmin}_{\mathbf{z}} f_2(\mathbf{z}) = \eta,$$

and

$$\inf_{\mathbf{z}} f_2(\mathbf{z}) = -\frac{1}{2}\|\eta\|_2^2.$$

Combining everything above, we get the dual problem:

$$\begin{aligned} \sup_{\eta} \quad & g(\eta) = \eta^T\mathbf{y} - \frac{1}{2}\|\eta\|_2^2, \\ \text{subject to} \quad & |\mathbf{x}_i^T\eta| \leq \lambda,\ i = 1, 2, \ldots, p. \end{aligned}$$

which is equivalent to

$$\begin{aligned} \sup_{\eta} \quad & g(\eta) = \frac{1}{2}\|\mathbf{y}\|_2^2 - \frac{1}{2}\|\eta - \mathbf{y}\|_2^2, \\ \text{subject to} \quad & |\mathbf{x}_i^T \eta| \leq \lambda,\ i = 1, 2, \ldots, p. \end{aligned} \tag{68}$$

By a simple re-scaling of the dual variables η, i.e., let $\theta = \frac{\eta}{\lambda}$, problem (68) transforms to:

$$\begin{aligned} \sup_{\theta} \quad & g(\theta) = \frac{1}{2}\|\mathbf{y}\|_2^2 - \frac{\lambda^2}{2}\|\theta - \frac{\mathbf{y}}{\lambda}\|_2^2, \\ \text{subject to} \quad & |\mathbf{x}_i^T \theta| \leq 1,\ i = 1, 2, \ldots, p. \end{aligned}$$

A.2 The KKT Conditions

Problem (64) is clearly convex and its constraints are all affine. By Slater's condition, as long as problem (64) is feasible we will have strong duality. Denote β^*, $\mathbf{z}^*$ and θ^* as optimal primal and dual variables. The Lagrangian is

$$L(\beta, \mathbf{z}, \theta) = \frac{1}{2}\|\mathbf{z}\|_2^2 + \lambda\|\beta\|_1 + \lambda\theta^T \cdot (\mathbf{y} - \mathbf{X}\beta - \mathbf{z}).$$

From the KKT condition, we have

$$0 \in \partial_\beta L(\beta^*, \mathbf{z}^*, \theta^*) = -\lambda \mathbf{X}^T \theta^* + \lambda \mathbf{v}, \text{ in which } \|\mathbf{v}\|_\infty \leq 1 \text{ and } \mathbf{v}^T \beta^* = \|\beta^*\|_1, \tag{69}$$

$$\nabla_{\mathbf{z}} L(\beta^*, \mathbf{z}^*, \theta^*) = \mathbf{z}^* - \lambda\theta^* = 0, \tag{70}$$

$$\nabla_\theta L(\beta^*, \mathbf{z}^*, \theta^*) = \lambda(\mathbf{y} - \mathbf{X}\beta^* - \mathbf{z}^*) = 0. \tag{71}$$

From (70) and (71), we have:

$$\mathbf{y} = \mathbf{X}\beta^* + \lambda\theta^*.$$

From (69), we know there exists $\mathbf{v}^* \in \partial\|\beta^*\|_1$ such that

$$\mathbf{X}^T\theta^* = \mathbf{v}^*,\ \|\mathbf{v}^*\|_\infty \leq 1 \text{ and } (\mathbf{v}^*)^T\beta^* = \|\beta^*\|_1,$$

which is equivalent to

$$|\mathbf{x}_i^T\theta^*| \leq 1, i = 1, 2, \ldots, p, \text{ and } (\theta^*)^T\mathbf{X}\beta^* = \|\beta^*\|_1. \tag{72}$$

From (72), it is easy to conclude:

$$(\theta^*)^T\mathbf{x}_i \in \begin{cases} \text{sign}(\beta_i^*) & \text{if } \beta_i^* \neq 0, \\ [-1, 1] & \text{if } \beta_i^* = 0. \end{cases}$$

Appendix B. The Dual Problem of Group Lasso

In this appendix, we present the detailed derivation of the dual problem of group Lasso.

B.1 Dual Formulation

Assuming the data matrix is $\mathbf{X}_g \in \mathbb{R}^{N \times n_g}$ and $p = \sum_{g=1}^{G} n_g$, the group Lasso problem is given by:

$$\inf_{\beta \in \mathbb{R}^p} \frac{1}{2}\|\mathbf{y} - \sum_{g=1}^{G} \mathbf{X}_g \beta_g\|_2^2 + \lambda \sum_{g=1}^{G} \sqrt{n_g}\|\beta_g\|_2. \tag{73}$$

Let $\mathbf{z} = \mathbf{y} - \sum_{g=1}^{G} \mathbf{X}_g \beta_g$ and problem (73) becomes:

$$\begin{aligned} \inf_{\beta} \quad & \frac{1}{2}\|\mathbf{z}\|_2^2 + \lambda \sum_{g=1}^{G} \sqrt{n_g}\|\beta_g\|_2, \\ \text{subject to} \quad & \mathbf{z} = \mathbf{y} - \sum_{g=1}^{G} \mathbf{X}_g \beta_g. \end{aligned} \tag{74}$$

By introducing the dual variables $\eta \in \mathbb{R}^N$, the Lagrangian of problem (74) is:

$$L(\beta, \mathbf{z}, \eta) = \frac{1}{2}\|\mathbf{z}\|_2^2 + \lambda \sum_{g=1}^{G} \sqrt{n_g}\|\beta_g\|_2 + \eta^T \cdot (\mathbf{y} - \sum_{g=1}^{G} \mathbf{X}_g \beta_g - \mathbf{z}).$$

and the dual function $g(\eta)$ is:

$$\begin{aligned} g(\eta) &= \inf_{\beta, \mathbf{z}} L(\beta, \mathbf{z}, \eta) \\ &= \eta^T \mathbf{y} + \inf_{\beta} \Big(- \eta^T \sum_{g=1}^{G} \mathbf{X}_g \beta_g + \lambda \sum_{g=1}^{G} \sqrt{n_g}\|\beta_g\|_2 \Big) + \inf_{\mathbf{z}} \big(\frac{1}{2}\|\mathbf{z}\|_2^2 - \eta^T \mathbf{z}\big). \end{aligned}$$

In order to get $g(\eta)$, let us solve the following two optimization problems.

$$\inf_{\beta} -\eta^T \sum_{g=1}^{G} \mathbf{X}_g \beta_g + \lambda \sum_{g=1}^{G} \sqrt{n_g}\|\beta_g\|_2, \tag{75}$$

and

$$\inf_{\mathbf{z}} \frac{1}{2}\|\mathbf{z}\|_2^2 - \eta^T \mathbf{z}. \tag{76}$$

Let us first consider problem (75). Denote the objective function of problem (75) as

$$\hat{f}(\beta) = -\eta^T \sum_{g=1}^{G} \mathbf{X}_g \beta_g + \lambda \sum_{g=1}^{G} \sqrt{n_g}\|\beta_g\|_2,$$

Let

$$\hat{f}_g(\beta_g) = -\eta^T \mathbf{X}_g \beta_g + \lambda \sqrt{n_g}\|\beta_g\|_2, \qquad g = 1, 2, \ldots, G.$$

then we can split problem (75) into a set of subproblems. Clearly $\hat{f}_g(\beta_g)$ is convex but not smooth because it has a singular point at 0. Consider the subgradient of $\hat{f}_g$,

$$\partial \hat{f}_g(\beta_g) = -\mathbf{X}_g^T \eta + \lambda \sqrt{n_g} \mathbf{v}_g, \qquad g = 1, 2, \ldots, G,$$

where $\mathbf{v}_g$ is the subgradient of $\|\beta_g\|_2$:

$$\mathbf{v}_g \in \begin{cases} \frac{\beta_g}{\|\beta_g\|_2} & \text{if } \beta_g \neq 0, \\ \mathbf{u},\ \|\mathbf{u}\|_2 \leq 1 & \text{if } \beta_g = 0. \end{cases} \tag{77}$$

Let β'_g be the optimal solution of $\hat{f}_g$, then β'_g satisfy

$$\exists \mathbf{v}'_g \in \partial \|\beta'_g\|_2, \quad -\mathbf{X}_g^T \eta + \lambda \sqrt{n_g} \mathbf{v}'_g = 0.$$

If $\beta'_g = 0$, clearly, $\hat{f}_g(\beta'_g) = 0$. Otherwise, since $\lambda \sqrt{n_g} \mathbf{v}'_g = \mathbf{X}_g^T \eta$ and $\mathbf{v}'_g = \frac{\beta'_g}{\|\beta'_g\|_2}$, we have

$$\hat{f}_g(\beta'_g) = -\lambda \sqrt{n_g} \frac{(\beta'_g)^T}{\|\beta'_g\|_2} \beta'_g + \lambda \sqrt{n_g} \|\beta'_g\|_2 = 0.$$

All together, we can conclude the

$$\inf_{\beta_g} \hat{f}_g(\beta_g) = 0, \quad g = 1, 2, \ldots, G$$

and thus

$$\inf_{\beta} \hat{f}(\beta) = \inf_{\beta} \sum_{g=1}^{G} \hat{f}_g(\beta_g) = \sum_{g=1}^{G} \inf_{\beta_g} \hat{f}_g(\beta_g) = 0.$$

The second equality is due to the fact that β_g's are independent.

Note, from (77), it is easy to see $\|\mathbf{v}_g\|_2 \leq 1$. Since $\lambda \sqrt{n_g} \mathbf{v}'_g = \mathbf{X}_g^T \eta$, we get a constraint on η, i.e., η should satisfy:

$$\|\mathbf{X}_g^T \eta\|_2 \leq \lambda \sqrt{n_g}, \qquad g = 1, 2, \ldots, G.$$

Next, let us consider problem (76). Since problem (76) is exactly the same as problem (66), we conclude:

$$\mathbf{z}' = \operatorname*{argmin}_{\mathbf{z}} \frac{1}{2} \|\mathbf{z}\|_2^2 - \eta^T \mathbf{z} = \eta,$$

and

$$\inf_{\mathbf{z}} \frac{1}{2} \|\mathbf{z}\|_2^2 - \eta^T \mathbf{z} = -\frac{1}{2} \|\eta\|_2^2.$$

Therefore the dual function $g(\eta)$ is:

$$g(\eta) = \eta^T \mathbf{y} - \frac{1}{2} \|\eta\|_2^2.$$

Combining everything above, we get the dual formulation of the group Lasso:

$$\begin{aligned} \sup_{\eta} \quad & g(\eta) = \eta^T \mathbf{y} - \frac{1}{2} \|\eta\|_2^2, \\ \text{subject to} \quad & \|\mathbf{X}_g^T \eta\|_2 \leq \lambda \sqrt{n_g},\ g = 1, 2, \ldots, G. \end{aligned}$$

which is equivalent to

$$\begin{aligned} \sup_{\eta} \quad & g(\eta) = \frac{1}{2}\|\mathbf{y}\|_2^2 - \frac{1}{2}\|\eta - \mathbf{y}\|_2^2, \\ \text{subject to} \quad & \|\mathbf{X}_g^T \eta\|_2 \leq \lambda\sqrt{n_g},\ g = 1, 2, \ldots, G. \end{aligned} \tag{78}$$

By a simple re-scaling of the dual variables η, i.e., let $\theta = \frac{\eta}{\lambda}$, problem (78) transforms to:

$$\begin{aligned} \sup_{\theta} \quad & g(\theta) = \frac{1}{2}\|\mathbf{y}\|_2^2 - \frac{\lambda^2}{2}\|\theta - \frac{\mathbf{y}}{\lambda}\|_2^2, \\ \text{subject to} \quad & \|\mathbf{X}_g^T \theta\|_2 \leq \sqrt{n_g},\ g = 1, 2, \ldots, G. \end{aligned}$$

B.2 The KKT Conditions

Clearly, problem (74) is convex and its constraints are all affine. By Slater's condition, as long as problem (74) is feasible we will have strong duality. Denote β^*, $\mathbf{z}^*$ and θ^* as optimal primal and dual variables. The Lagrangian is

$$L(\beta, \mathbf{z}, \theta) = \frac{1}{2}\|\mathbf{z}\|_2^2 + \lambda \sum_{g=1}^{G} \sqrt{n_g}\|\beta_g\|_2 + \lambda\theta^T \cdot (\mathbf{y} - \sum_{g=1}^{G} \mathbf{X}_g\beta_g - \mathbf{z}).$$

From the KKT condition, we have

$$0 \in \partial_{\beta_g} L(\beta^*, \mathbf{z}^*, \theta^*) = -\lambda \mathbf{X}_g^T \theta^* + \lambda\sqrt{n_g}\mathbf{v}_g, \text{ in which } \mathbf{v}_g \in \partial\|\beta_g^*\|_2, \quad g = 1, 2, \ldots, G, \tag{79}$$

$$\nabla_{\mathbf{z}} L(\beta^*, \mathbf{z}^*, \theta^*) = \mathbf{z}^* - \lambda\theta^* = 0, \tag{80}$$

$$\nabla_{\theta} L(\beta^*, \mathbf{z}^*, \theta^*) = \lambda \cdot (\mathbf{y} - \sum_{g=1}^{G} \mathbf{X}_g\beta_g^* - \mathbf{z}^*) = 0. \tag{81}$$

From (80) and (81), we have:

$$\mathbf{y} = \sum_{g=1}^{G} \mathbf{X}_g\beta_g^* + \lambda\theta^*.$$

From (79), we know there exists $\mathbf{v}_g' \in \partial\|\beta_g^*\|_2$ such that

$$\mathbf{X}_g^T\theta^* = \sqrt{n_g}\mathbf{v}_g'$$

and

$$\mathbf{v}_g' \in \begin{cases} \frac{\beta_g^*}{\|\beta_g^*\|_2} & \text{if } \beta_g^* \neq 0, \\ \mathbf{u},\ \|\mathbf{u}\|_2 \leq 1 & \text{if } \beta_g^* = 0, \end{cases}$$

Then the following holds:

$$\mathbf{X}_g^T\theta^* \in \begin{cases} \sqrt{n_g}\frac{\beta_g^*}{\|\beta_g^*\|_2} & \text{if } \beta_g^* \neq 0, \\ \sqrt{n_g}\mathbf{u},\ \|\mathbf{u}\|_2 \leq 1 \text{ if } \beta_g^* = 0, \end{cases}$$

for $g = 1, 2, \ldots, G$. Clearly, if $\|\mathbf{X}_g^T\theta^*\|_2 < \sqrt{n_g}$, we can conclude $\beta_g^* = 0$.

References

U. Alon, N. Barkai, D. Notterman, K. Gish, S. Ybarra, D. Mack, and A. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Cell Biology*, 96:6745–6750, 1999.

S. Armstrong, J. Staunton, L. Silverman, R. Pieters, M. den Boer, M. Minden, S. Sallan, E. Lander, T. Golub, and S. Korsmeyer. MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia. *Nature Genetics*, 30:41–47, 2002.

H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer, 2011.

S. R. Becker, E. Candès, and M. Grant. Templates for convex cone problems with applications to sparse signal recovery. Technical report, Standford University, 2010.

D. P. Bertsekas. *Convex Analysis and Optimization*. Athena Scientific, 2003.

A. Bhattacharjee, W. Richards, J. Staunton, C. Li, S. Monti, P. Vasa, C. Ladd, J. Beheshti, R. Bueno, M. Gillette, M. Loda, G. Weber, E. Mark, E. Lander, W. Wong, B. Johnson, T. Golub, D. Sugarbaker, and M. Meyerson. Classification of human lung carcinomas by mrna expression profiling reveals distinct adenocarcinoma subclasses. *Proceedings of the National Academy of Sciences*, 98:13790–13795, 2001.

H. Bondell and B. Reich. Simultaneous regression shrinkage, variable selection and clustering of predictors with OSCAR. *Biometrics*, 64:115–123, 2008.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

A. Bruckstein, D. Donoho, and M. Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Review*, 51:34–81, 2009.

D. Cai, X. He, and J. Han. Efficient kernel discriminant analysis via spectral regression. In *ICDM*, 2007.

D. Cai, X. He, J. Han, and T. Huang. Graph regularized non-negative matrix factorization for data representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33:1548–1560, 2011.

E. Candès. Compressive sampling. In *Proceedings of the International Congress of Mathematics*, 2006.

S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Review*, 43:129–159, 2001.

D. L. Donoho and Y. Tsaig. Fast solution of l-1 norm minimization problems when the solution may be sparse. *IEEE Transactions on Information Theory*, 54:4789–4812, 2008.

B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32:407–499, 2004.

L. El Ghaoui, V. Viallon, and T. Rabbani. Safe feature elimination in sparse supervised learning. *Pacific Journal of Optimization*, 8:667–698, 2012.

J. Fan and J. Lv. Sure independence screening for ultrahigh dimensional feature spaces. *Journal of the Royal Statistical Society Series B*, 70:849–911, 2008.

J. Friedman, T. Hastie, H. Hëfling, and R. Tibshirani. Pathwise coordinate optimization. *Annals of Applied Statistics*, 1:302–332, 2007.

J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33:1–22, 2010.

S. J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. An interior-point method for large scale l1-regularized least squares. *IEEE Journal on Selected Topics in Signal Processing*, 1:606–617, 2007.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, 1998.

J. Liu, S. Ji, and J. Ye. *SLEP: Sparse Learning with Efficient Projections*. Arizona State University, 2009.

J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11:19–60, 2010.

S. Nene, S. Nayar, and H. Murase. Columbia object image library (coil-100). Technical report, CUCS-006-96, Columbia University, 1996.

Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Ng. Reading digits in nature images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2001.

M. Y. Park and T. Hastie. L1-regularized path algorithm for generalized linear models. *Journal of the Royal Statistical Society Series B*, 69:659–677, 2007.

E. Petricoin, D. Ornstein, C. Paweletz, A. Ardekani, P. Hackett, B. Hitt, A. Velassco, C. Trucco, L. Wiegand, K. Wood, C. Simone, P. Levine, W. Linehan, M. Emmert-Buck, S. Steinberg, E. Kohn, and L. Liotta. Serum proteomic patterns for detection of prostate cancer. *Journal of National Cancer Institute*, 94:1576–1578, 2002.

A. Ruszczyński. *Nonlinear Optimization*. Princeton University Press, 2006.

S. Shevade and S. Keerthi. A simple and efficient algorithm for gene selection using sparse logistic regression. *Bioinformatics*, 19:2246–2253, 2003.

T. Sim, B. Baker, and M. Bsat. The CMU pose, illumination, and expression database. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:1615–1618, 2003.

R. Tibshirani. Regression shringkage and selection via the lasso. *Journal of the Royal Statistical Society Series B*, 58:267–288, 1996.

R. Tibshirani, J. Bien, J. Friedman, T. Hastie, N. Simon, J. Taylor, and R. Tibshirani. Strong rules for discarding predictors in lasso-type problems. *Journal of the Royal Statistical Society Series B*, 74:245–266, 2012.

M. West, C. Blanchette, H. Dressman, E. Huang, S. Ishida, R. Spang, H. Zuzan, J. Olson, J. Marks, and J. Nevins. Predicting the clinical status of human breast cancer by using gene expression profiles. *Proceedings of the National Academy of Sciences*, 98:11462–11467, 2001.

J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. Huang, and S. Yan. Sparse representation for computer vision and pattern recognition. In *Proceedings of IEEE*, 2010.

Z. J. Xiang, H. Xu, and P. J. Ramadge. Learning sparse representation of high dimensional data on large scale dictionaries. In *NIPS*, 2011.

M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society Series B*, 68:49–67, 2006.

P. Zhao and B. Yu. On model selection consistency of lasso. *Journal of Machine Learning Research*, 7:2541–2563, 2006.

H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B*, 67:301–320, 2005.

Journal of Machine Learning Research 16 (2015) 1103-1155 Submitted 10/14; Revised 11/14; Published 6/15

Fast Cross-Validation via Sequential Testing

Tammo Krueger TAMMOKRUEGER@GMAIL.COM
Danny Panknin PANKNIN@CS.TU-BERLIN.DE
Mikio Braun MIKIO.BRAUN@TU-BERLIN.DE
Technische Universität Berlin
Machine Learning Group
Marchstr. 23, MAR 4-1
10587 Berlin, Germany

Editor: Charles Elkan

Abstract

With the increasing size of today's data sets, finding the right parameter configuration in model selection via cross-validation can be an extremely time-consuming task. In this paper we propose an improved cross-validation procedure which uses nonparametric testing coupled with sequential analysis to determine the best parameter set on linearly increasing subsets of the data. By eliminating underperforming candidates quickly and keeping promising candidates as long as possible, the method speeds up the computation while preserving the power of the full cross-validation. Theoretical considerations underline the statistical power of our procedure. The experimental evaluation shows that our method reduces the computation time by a factor of up to 120 compared to a full cross-validation with a negligible impact on the accuracy.

Keywords: cross-validation, statistical testing, nonparametric methods

1. Introduction

Model selection by cross-validation is a de-facto standard in applied machine learning to tune parameter configurations of machine learning methods in supervised learning settings (see Mosteller and Tukey 1968; Stone 1974; Geisser 1975 and also Arlot et al. 2010 for a recent and extensive review of the method). Part of the data is held back and used as a test set to get a less biased estimate of the true generalization error. Cross-validation is computationally quite demanding, though. Doing a full grid search on all possible combinations of parameter candidates quickly takes a lot of time, even if one exploits the obvious potential for parallelization.

Therefore, cross-validation is seldom executed in full in practice, but different heuristics are usually employed to speed up the computation. For example, instead of using the full grid, local search heuristics may be used to find local minima in the test error (see for instance Kohavi and John 1995; Bengio 2000; Keerthi et al. 2006). However, in general, as with all local search methods, no guarantees can be given as to the quality of the found local minima. Another frequently used heuristic is to perform the cross-validation on a subset of the data, and then train on the full data set to get the most accurate predictions. The problem here is to find the right size of the subset: If the subset is too small and

cannot reflect the true complexity of the learning problem, the configurations selected by cross-validation will lead to underfitted models. On the other hand, a too large subset will take longer for the cross-validation to finish.

Effective use of model selection heuristics requires both an experienced practitioner and familiarity with the data set. However, as we will discuss in more depth below, the effect of taking subsets on the estimated generalization error is more manageable: Given increasing subsets of the data, the test errors converge to the values on the full data set for each parameter configuration, but the parameter configuration achieving the minimum test error will converge much faster. Thus, using subsets in a systematic way opens up a promising way to speed up the model selection process, since training models on smaller subsets of the data is much more time-efficient. During this process care has to be taken when an increase in available data suddenly reveals more structure in the data, leading to a change of the optimal parameter configuration. Still, as we will discuss in more depth, there are ways to guard against such change points, making the heuristic of taking subsets a more promising candidate for an automated procedure.

In this paper we will propose a method which speeds up cross-validation by considering subsets of increasing size. By removing clearly underperforming parameter configurations on the way this leads to a substantial saving in total computation time as sketched in Figure 1. In order to account for possible change points, sequential testing (Wald, 1947) is adapted to control a *safety zone*, roughly speaking, a certain number of allowed failures for a parameter configuration; at the same time this framework allows for dropping clearly underperforming configurations. Finally, we add a stopping criterion to watch for early convergence of the process to further speed up the computation. The resulting method thus consumes less time and space than a full grid cross-validation procedure at no significant loss in accuracy. We prove certain theoretical properties about its optimality, yet, this procedure relies on the availability of a vast amount of data to guide the decision process into a stable region where each configuration sees enough data to show its real performance.

In the following, we will first discuss the effects of taking subsets on learners and cross-validation (Section 2), discuss related work in Section 3, present our method Fast Cross-Validation via Sequential Testing (CVST, Section 4), state the theoretical properties of the method (Section 5) and finally evaluate our method on synthetic and real-world data sets in Section 6. Section 7 gives an overview of possible extensions and Section 8 concludes the paper. The impatient practitioner may skip some theoretical treatments and focus on the self-contained Section 4 describing the CVST algorithm and its evaluation in Section 6. To ease the reading process we collected our notational conventions in Table 1.

2. Cross-Validation on Subsets

Our approach is based on taking subsets of the data to speed up cross-validation. For this approach to work well, we need that the minima of the test errors give reliable estimates of the true performance already for small subsets. In this section, we will discuss the setting and motivate our approach. The goal is to understand which effects lead to making the estimates reliable.

Let us first introduce some notation: Assume that our N training data points d_i are given by input/output pairs $d_i = (X_i, Y_i) \in \mathcal{X} \times \mathcal{Y}$ drawn i.i.d. from some probability

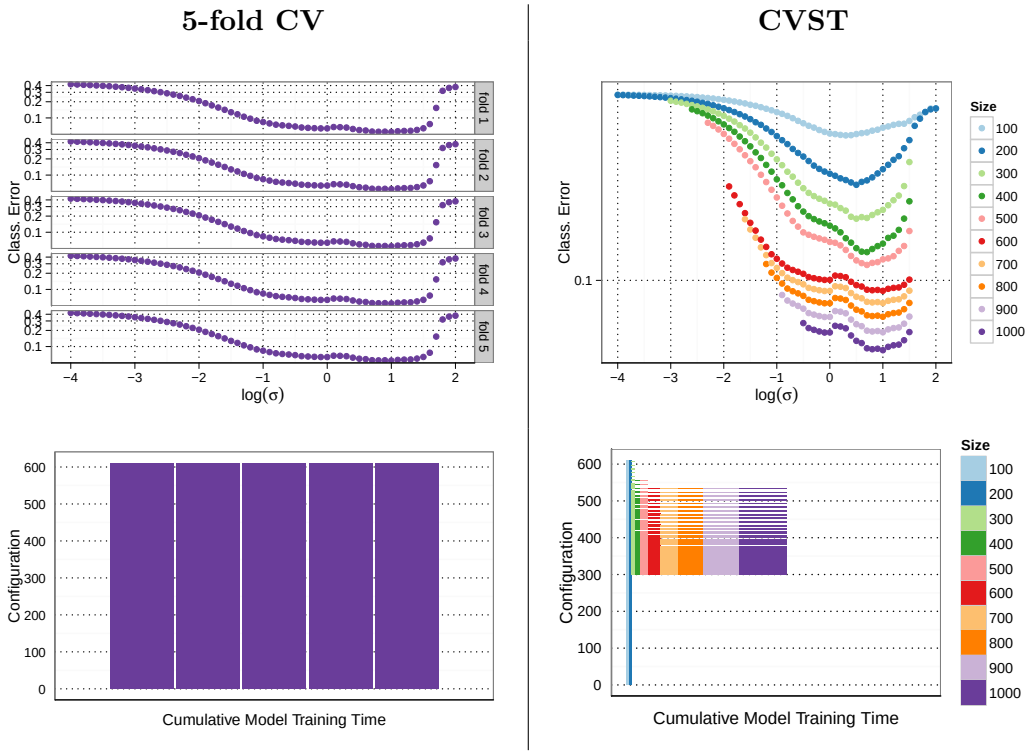


Figure 1: Performance of a 5-fold cross-validation (CV, left) and fast cross-validation via sequential testing (CVST, right): While the CV has to calculate the model for each configuration (here: σ of a Gaussian kernel) on the full data set, the CVST algorithm uses increasing subsets of the data and drops significantly underperforming configurations in each step (upper panels), resulting in a drastic decrease of total calculation time (sum of colored area in lower panels).

distribution P on $\mathcal{X} \times \mathcal{Y}$. We assume an example-wise loss function $\ell\colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ so that the overall error or expected risk of a predictor $g\colon \mathcal{X} \to \mathcal{Y}$ is given by $R(g) = E[\ell(g(X), Y)]$ where $(X, Y) \sim P$. For some finite set of possible parameter configurations C, let $g_n(c)$ be the predictor learned for parameter $c \in C$ from the first n training examples.

The core procedure in a cross-validation approach is to train predictors for each c and consider their test error. Denote by $g_n(c)$ the predictor obtained by training on the first n points of the training data for parameter c. We wish to study whether this error converges as n grows. Let us denote by c_n^* a configuration optimal for subset size n:

$$R(g_n(c_n^*)) = \min_{c \in C} R(g_n(c)).$$

We will ignore cases where there are multiple minima c_n^*, because we are only interested in the test error achieved, not the location of the minimum.

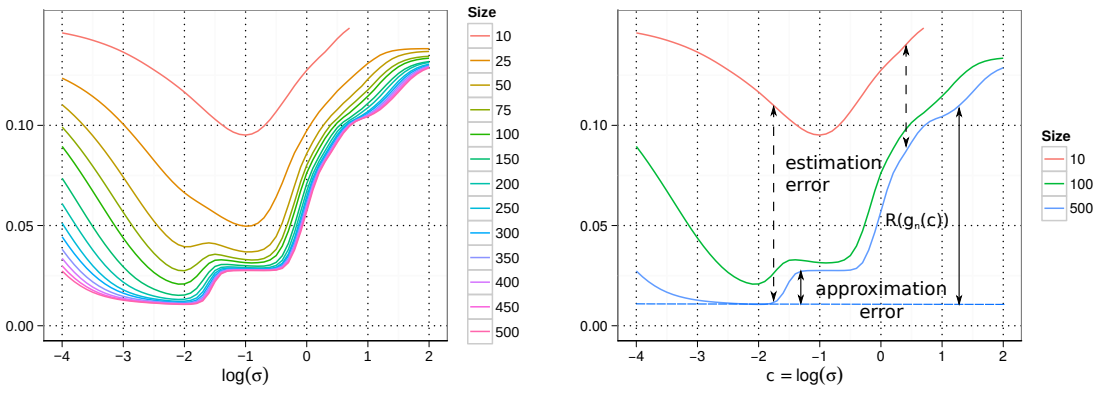
Symbol	Description
$d_i = (X_i, Y_i) \in \mathcal{X} \times \mathcal{Y}$	Data points
N	Total data set size
$g : \mathcal{X} \mapsto \mathcal{Y}$	Learned predictor
$\ell : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}$	Loss function
$R(g) = E[\ell(g(X), Y)]$	Risk of predictor g
c	Configuration of learner
C	Finite set of examined configurations
$g_n(c) : \mathcal{X} \mapsto \mathcal{Y}$	Predictor learned on n data points for configuration c
c^*	Overall best configuration
c_n^*	Best configuration for models based on n data points
s	Current step of CVST procedure
S	Total number of steps
$\Delta = N/S$	Increment of model size
P_p	Pointwise performance matrix
P_S	Overall performance matrix of dimension $\lvert C\rvert \times S$
T_S	Trace matrix of dimension $\lvert C\rvert \times S$
w_{stop}	Size of early stopping window
$\alpha, \alpha_l, \beta_l$	Significance levels
π	Success probability of a binomial variable

Table 1: List of symbols

In cross-validation, the true test error is not known and estimated by the empirical error on an independent test set. For the sake of simplicity, we will consider the true test error nevertheless for the remainder of this section. In our experience, the effects discussed below also hold for cross-validation, because the estimation error is small and does not create a systematic distortion of the choice of configuration.

Since we want to infer the performance of the predictor on the full training set based on its performance on a subset, we need that the errors are similar for a fixed configuration c as the size of the subset approaches the full training set size. A necessary condition for this to hold in general is that $R(g_n(c))$ converges as n tends to infinity. Luckily, this holds for most existing learning methods (see Appendix A for some examples). A counter example is the case of k-nearest neighbor with fixed k. Training with $k = 10$ leads to quite different predictions on data sets of size 100 compared to, say, 10,000. More discussion can be found below in Section 5.3.

We are interested in the difference in errors between the best parameter configuration learned on the subset of size n, and on the full data set N, that is, $R(g_n(c_n^*)) - R(g_N(c_N^*))$. This error can be bounded by considering the difference between $R(g_n(c))$ and $R(g_N(c))$ uniformly over the whole configuration set C. If the learner itself converges it is trivial to show that the errors also converge for finite parameter configuration sets C. On the other hand, uniform convergence is quite a strong requirement, since it requires that the test errors also converge for suboptimal configurations. In particular for parameter configurations c



(a) (b)

Figure 2: Test error of an SVR model on the *noisy sinc* data set introduced in Section 6.1. We can observe a shift of the optimal σ of the Gaussian kernel to the fine-grained structure of the problem, if we have seen enough data. In Figure (b), approximation error is indicated by the black solid line, and the estimation error by the black dashed line. The minimal risk is shown as the blue dashed line. One can see that uniform approximation of the estimation error is not the main driving force, instead, the decay of the approximation error with smaller kernel widths together with an increase of the estimation error at small kernel widths makes sure that the minimum converges quickly.

which correspond to complex models, $g_n(c)$ may continue to improve right up to the full number N of data points.

So while uniform convergence seems a sufficient condition, let us look at a concrete example to see whether uniform convergence is a necessary condition for convergence of the minima. Figure 2(a) shows the test errors for a typical example. We train a support vector regression model (SVR) on subsets of the full training set consisting of 500 data points. The data set is the *noisy sinc* data set introduced in Section 6.1. Model parameters are the kernel width σ of the Gaussian kernel used and the regularization parameter, where the results shown are already optimized over the regularization parameter for the sake of simplicity.

We see that the minimum converges rather quickly, first to the plateau of $\log(\sigma) \in [-1.5, -0.3]$ approximately, and then towards the lower one at $[-2.5, -1.7]$, which is also the optimal one at training set size $n = 500$. We see that uniform convergence is not the main driving force. In fact, the errors for small kernel widths are still very far apart even when the minimum is already converged.

In the following, it is helpful to continue the discussion using the concepts of estimation error and approximation error. We assume that the learner is trained by picking the model which minimizes the empirical risk over some hypothesis set $\mathcal{G}$. Let us denote this predictor as g_n^*. In this setting, one can write the difference between the expected risk of the predictor $R(g_n^*)$ and the Bayes risk R^* as follows (see Section 12.1 in Devroye et al. 1996 or

Section 2.4.3 in Mohri et al. 2012):

$$R(g_n^*) - R^* = \underbrace{\left(R(g_n^*) - \inf_{g \in \mathcal{G}} R(g)\right)}_{\text{estimation error}} + \underbrace{\left(\inf_{g \in \mathcal{G}} R(g) - R^*\right)}_{\text{approximation error}}.$$

The estimation error measures how far the chosen model is from the one which would be asymptotically optimal, while the approximation error measures the difference in risk between the best possible model in the hypothesis class and the true function.

Using this decomposition, we can interpret the figure as follows (see Figure 2(b)): The kernel width controls the approximation error. For $\log(\sigma) \geq -1.8$, the resulting hypothesis class is too coarse to represent the function under consideration. It becomes smaller until it reaches the level of the Bayes risk as indicated by the dashed blue line. For even larger training set sizes, we can assume that it will stay on this level even for smaller kernel sizes.

The difference between the blue line and the upper lines shows the estimation error. The estimation error has been extensively studied in statistical learning theory and is known to be linked to different notions of complexity like VC-dimension (Vapnik, 1998), fat-shattering dimension (Bartlett et al., 1996), or the norm in the reproducing kernel Hilbert space (RKHS) (Evgeniou and Pontil, 1999). A typical result shows that the estimation error can be bounded by terms of the form

$$R(g_n^*) - \inf_{g \in \mathcal{G}} R(g) \leq O\left(\sqrt{\frac{d(\mathcal{G}) \log n}{n}}\right),$$

where $d(\mathcal{G})$ is some notion of complexity of the underlying hypothesis class, and the bound holds with high probability. For our figure, this means that we can expect the estimation error to become larger for smaller kernel widths.

If we image the parameter configurations ordered according to their complexity, we see that for parameter configurations with small complexity (that is, large kernel width), the approximation error will be high, but the estimation error will be small. At the same time, for parameter configurations with high complexity, the approximation error will be small, even optimal, but the estimation error will be large, although it will decay with increasing training set size. In combination, the estimates at smaller training set sizes tend to underestimate the true model complexity, but as the estimation error decreases and becomes small compared to the approximation error, the minimum also converges to the true one. The fact that the estimation error is larger for more complex models acts as a guard to choose too complex models. The estimation error for models which have higher complexity than the optimal one can effectively be ignored. Therefore, we can expect much faster convergence than given by a uniform error bound, which is, however, highly data dependent.

Unfortunately, existing theoretical results are not able to bound the error sufficiently tightly to make these arguments more exact. In particular, the speed of the convergence on the minimum hinges on a tight lower bound on the approximation error, and a realistic upper bound on the estimation error. Approximation errors have been studied for example in the papers by Smale and Zhou (2003) and Steinwart and Scovel (2007), but the papers only prove upper bounds, and the rates are also worst-case rates which are likely not close

enough to the true errors. A more formal study of the effects discussed above is therefore the subject of future work.

On the other hand, the mechanisms which lead to fast convergence of the minimum are plausible when looking at concrete examples as we did above. Therefore, we will assume in the following that the location of the best parameter configuration might initially change but then become more or less stable quickly. Note that we do not claim that the speed of this convergence is known. Instead, we will use sequential testing to introduce a *safety zone* which will be as large as possible to ensure that our method is robust against these initial changes and good configurations survive till final stable regime.

3. Related Work

Using statistical tests and the sequential analysis framework in order to speed up learning has been the topic of several lines of research. However, the existing body of work mostly focuses on reducing the number of test evaluations, while we focus on the overall process of eliminating candidates themselves. To the best of our knowledge, this is a new concept and can apparently be combined with the already available racing techniques to further reduce the total calculation time.

Maron and Moore (1994, 1997) introduce the so-called *Hoeffding Races* which are based on the nonparametric Hoeffding bound for the mean of the test error. At each step of the algorithm a new test point is evaluated by all remaining models and the confidence intervals of the test errors are updated accordingly. Models whose confidence interval of the test error lies outside of at least one interval of a better performing model are dropped. In a similar vein Zheng and Bilenko (2013) have applied this concept to cross-validation and improve this approach by using paired t-test and power analysis to control both the false positive and false negative rate. Chien et al. (1995, 1999) devise a similar range of algorithms using concepts of PAC learning and game theory: Different hypotheses are ordered by their expected utility according to the test data the algorithm has seen so far. As for Hoeffding Races, the emphasis in this approach lies on reducing the number of evaluations. Thus, the application domain for these kind of algorithms is best suited where the evaluation of a data point given a learned model is costly. Since this approach expects that a model is fully trained before its evaluation, the direct utilization of racing algorithms for model selection would result in a procedure similar to a one-fold cross-validation: First learn a model on one half of the data and do the time efficient evaluation as described above on the other half. Obviously, this would yield a maximal relative time improvement of k compared to standard k-fold cross-validation since we learn one model instead of the k for k-fold cross-validation. Yet, the orthogonality of this approach to the CVST procedure could be utilized in each step and for each remaining configuration to further increase the runtime benefits by minimizing the necessary evaluations of a model for determining whether it belongs to the top configurations or not.

This concept of racing is further extended by Domingos and Hulten (2001): By introducing an upper bound for the learner's loss as a function of the examples, the procedure allows for an early stopping of the learning process, if the loss is nearly as optimal as for infinite data. Birattari et al. (2002) apply racing in the domain of evolutionary algorithms and extend the framework by using the Friedman test to filter out non-promising configu-

rations. While Bradley and Schapire (2008) use similar concepts in the context of boosting (FilterBoost), Mnih et al. (2008) introduce the empirical Bernstein Bounds to extend both the FilterBoost framework and the racing algorithms. In both cases the bounds are used to estimate the error within a specific ϵ-region with a given probability. Pelossof and Jones (2009) use the concept of sequential testing to speed up the boosting process by controlling the number of features which are evaluated for each sample. In a similar fashion this approach is used in Pelossof and Ying (2010) to increase the speed of the evaluation of the perceptron and in Pelossof and Ying (2011) to speed up the Pegasos algorithm. Stanski (2012) uses a partial leave-one-out evaluation of model performance to get an estimate of the overall model performance, which is used to pick the most probable best model. These racing concepts are applied in a wide variety of domains like reinforcement learning (Heidrich-Meisner and Igel, 2009) and timetabling (Birattari, 2009) showing the relevance and practical impact of the topic.

Recently, Bayesian optimization has been applied to the problem of hyper-parameter optimization of machine learning algorithms. Bergstra et al. (2011) use the sequential model-based global optimization framework (SMBO) and implement the loss function of an algorithm via hierarchical Gaussian processes. Given the previously observed history of performances, a candidate configuration is selected which minimizes this historical surrogate loss function. Applied to the problem of training deep belief networks this approach shows superior performance over random search strategies. Snoek et al. (2012) extend this approach by including timing information for each potential model, i.e., the cost of learning a model and optimizing the expected improvement per seconds leads to a global optimization in terms of wall-clock time. Thornton et al. (2012) apply the SMBO framework in the context of the WEKA machine learning toolbox: The so-called Auto-WEKA procedure not only finds the optimal parameter for a specific learning problem but also searches for the most suitable learning algorithm. Like the racing concepts, these Bayesian optimization approaches are orthogonal to the CVST approach and could be combined to speed up each step of the CVST loop.

On first sight, the multi-armed bandit problem (Berry and Fristedt, 1985; Cesa-Bianchi and Lugosi, 2006) also seems to be related to the problem here in another way: In the multi-armed bandit problem, a number of distributions are given and the task is to identify the distribution with the largest mean from a chosen sequence of samples from the individual distributions. In each round, the agent chooses one distribution to sample from and typically has to find some balance between exploring the different distributions, rejecting distributions which do not seem promising and focusing on a few candidates to get more accurate samples.

This looks similar to our setting where we also wish to identify promising candidates and reject underperforming configurations early on in the process, but the main difference is that the multi-armed bandit setting assumes that the distributions are fixed whereas we specifically have to deal with distributions which change as the sample size increases. This leads to the introduction of a safety zone, among other things. Therefore, the multi-armed bandit setting is not applicable across different sample sizes. On the other hand, the multi-armed bandit approach is a possible extension to speed up the computation within a fixed training size similar to the Hoeffding races already mentioned above.

4. Fast Cross-Validation via Sequential Testing (CVST)

Recall from Section 2 that we have a data set consisting of N data points $d_i = (X_i, Y_i) \in \mathcal{X} \times \mathcal{Y}$ which we assume to be drawn i.i.d. from P. We have a learning algorithm which depends on several parameters collected in a configuration $c \in C$. The goal is to select the configuration c^* out of all possible configurations C such that the learned predictor g has the best generalization error with respect to some loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$.

Our approach attempts to speed up the model selection process by learning just on subsamples of size $n := s\frac{N}{S} = s\Delta$ for $1 \leq s \leq S$ where S is the maximal number of steps the CVST algorithm should run. The procedure starts with the full set of configurations and eliminates clearly underperforming configurations at each step s based on the performances observed in steps 1 to s. The main loop of Algorithm 1 on page 1112 executes the following parts at each step s:

❶ The procedure learns a model on the first n data points for the remaining configurations and stores the test errors on the remaining $N - n$ data points in the pointwise performance matrix P_p (Lines 10-14). This matrix P_p is used on Lines 15-16 to estimate the top performing configurations via robust testing (see Algorithm 2) and saves the outcome as a binary "top or flop" scheme accordingly.

❷ The procedure drops significant loser configurations along the way (Lines 17-19 and Algorithm 3) using tests from the sequential analysis framework.

❸ Applying robust, distribution free testing techniques allows for an early stopping of the procedure when we have seen enough data for a stable parameter estimation (Line 20 and Algorithm 4).

In the following we will discuss the individual steps in the algorithm and formally define the notations used. A conceptual overview of one iteration of the procedure is depicted in Figure 3 for reference. Additionally, we have released a software package on CRAN named `CVST` which is publicly available via all official CRAN repositories and also via GitHub (`https://github.com/tammok/CVST`). This package contains the CVST procedure and all learners used in Section 6 ready for use.

4.1 Robust Transformation of Test Errors

To robustly transform the performance of configurations into the binary information whether it is among the top-performing configurations or turns out to be a flop, we rely on distribution-free tests. The basic idea is to calculate the performance of a given configuration on data points not used during learning and store this information in the pointwise performance matrix P_p. Then we find the group of best configurations by first ordering them according to their mean performance in this step and then compare in a stepwise fashion whether the pointwise performance matrix P_p of a given subset of the configurations are significantly different.

We give an example of this procedure by the situation depicted in Figure 3 with K remaining configurations $c_1, c_2, \ldots, c_K$ which are ordered according to their mean performances (i.e., sorted ascending with regard to their expected loss). We now want to find the

Algorithm 1 CVST Main Loop

function CVST($d_1, \ldots, d_N$, S, C, α, β_l, α_l, w_{stop})
  $\Delta \leftarrow \text{N}/S$ ▷ Initialize subset increment
  $n \leftarrow \Delta$ ▷ Initialize model size
  test $\leftarrow$getTest(S, β_l, α_l) ▷ Get sequential test
  $\forall \text{s} \in \{1, \ldots, S\}, c \in C : T_S[c, s] \leftarrow 0$
  $\forall \text{s} \in \{1, \ldots, S\}, c \in C : P_S[c, s] \leftarrow \text{NA}$
  $\forall c \in C : \text{isActive}[c] \leftarrow$**true**
  for $s \leftarrow 1$ **to** S **do**
    $\forall \text{i} \in \{1, \ldots, N - n\}, c \in C : P_p[c, i] \leftarrow \text{NA}$
    for $c \in C$ **do**
      if isActive[c] **then**
        $g = g_n(c)$ ▷ Learn model on the first n data points
        $\forall i \in \{1, \ldots, N - n\} : P_p[c, i] \leftarrow \ell(g(x_{n+i}), y_{n+i})$ ▷ Evaluate on the rest
        $P_S[c, s] \leftarrow \frac{1}{N-n} \sum_{i=1}^{N-n} P_p[c, i]$ ▷ Store mean performance
    $\text{index}_{\text{top}} \leftarrow$topConfigurations($P_p, \alpha$) ▷ Find the top configurations
    $T_S[\text{index}_{\text{top}}, s] \leftarrow 1$ ▷ And set entry in trace matrix
    for $c \in C$ **do**
      if isActive[c] and isFlopConfiguration($T_S[c, 1 : s]$, s, S, β_l, α_l) **then**
        isActive[c] $\leftarrow$**false** ▷ De-activate flop configuration
    if similarPerformance(T_S[isActive, $(s - w_{\text{stop}} + 1) : s$], α) **then**
      break
    $n \leftarrow n + \Delta$
  return selectWinnner(P_S, isActive, w_{stop}, s)

Algorithm 2 Find the top configurations via iterative testing

function topConfigurations(P_p, α)
  $\forall i \in \{1, \ldots, C\} : P_m[k] \leftarrow \frac{1}{N-n} \sum_{j=1}^{N-n} P_p[k, j]$
  $\text{index}_{\text{sort}} \leftarrow$ sortIndexDecreasing(P_m)
  $\widetilde{P}_p = P_p[\text{index}_{\text{sort}},]$ ▷ Sort P_p according to the mean performance
  $K \leftarrow$which(isNA(P_m)) $- 1$ ▷ K is the number of active configurations
  $\tilde{\alpha} = \alpha/(K - 1)$ ▷ Bonferroni correction for $K - 1$ potential tests
  for $k \in \{2, \ldots, K\}$ **do**
    if is classification task **then** ▷ Choose according test
      $p \leftarrow$ cochranQTest($\widetilde{P}_p[1 : k,]$)
    else
      $p \leftarrow$ friedmanTest($\widetilde{P}_p[1 : k,]$)
    if $p \leq \tilde{\alpha}$ **then** ▷ We found a significant effect
      break ▷ so the $k - 1^{\text{th}}$ preceding configurations are the top ones
  return $\text{index}_{\text{sort}}[1 : (k - 1)]$

Algorithm 3 Check for flop configurations via sequential testing

function isFlopConfiguration(T, s, S, β_l, α_l)
  $\pi_0 \leftarrow 0.5; \pi_1 \leftarrow \frac{1}{2}\sqrt[S]{\frac{1-\beta_l}{\alpha_l}}$
  $a \leftarrow \frac{\log\frac{\beta_l}{1-\alpha_l}}{\log\frac{\pi_1}{\pi_0}-\log\frac{1-\pi_1}{1-\pi_0}}$
  $b \leftarrow \frac{\log\frac{1-\pi_0}{1-\pi_1}}{\log\frac{\pi_1}{\pi_0}-\log\frac{1-\pi_1}{1-\pi_0}}$
  return $\sum_{i=1}^{s} T_i \leq a + bs$

Algorithm 4 Compare performance of remaining configurations

function similarPerformance(T_S, α)
  $p \leftarrow$ cochranQTest(T_S)
  return $p \leq \alpha$

smallest index $k \leq K$, such that the configurations $c_1, c_2, \ldots, c_k$ all show a similar behavior on the remaining data points $d_{n+1}, d_{n+2}, \ldots, d_N$ not used in the current model learning process based on a statistical test.

The rationale behind our comparison procedure is three-fold: First, by ordering the configurations by the mean performances we start with the comparison of the currently best performing configurations first. Second, by using the first $n := s\Delta$ data points for the model building and the remaining $N - n$ data points for the estimation of the average performance of each configuration, we compensate the error introduced by learning on smaller subsets of the data by better error estimates on more data points. I.e., for small s we will learn the model on relatively small subsets of the overall available data while we estimate the test error on relatively large portions of the data and vice versa. Third, by applying test procedures directly on the error estimates of individual data points we exploit a further robustifying pooling effect: If we have outliers in the testing data, all models will be affected by this and therefore the overall testing result will not be affected. We will

Algorithm 5 Select the winning configuration out of the remaining ones

function selectWinnner(P_S, isActive, w_{stop}, s)
  $\forall \text{i} \in \{1, \ldots, s\}, c \in C : R_S[c, i] \leftarrow \infty$
  for $i \in \{1, \ldots, s\}$ **do**
    for $c \in C$ **do**
      if isActive[c] **then**
        $R_S[c, i] = \text{rank}(P_S[c, i], P_S[, i])$ ▷ Gather the rank of c in step i
  $\forall c \in C : M_S[c] \leftarrow \infty$
  for $c \in C$ **do**
    if isActive[c] **then**
      $M_S[c] \leftarrow \frac{1}{w_{\text{stop}}} \sum_{i=s-w_{\text{stop}}+1}^{s} R_S[c, i]$ ▷ Mean rank for the last w_{stop} steps
  return whichMin(M_S) ▷ Return configuration with minimal mean rank

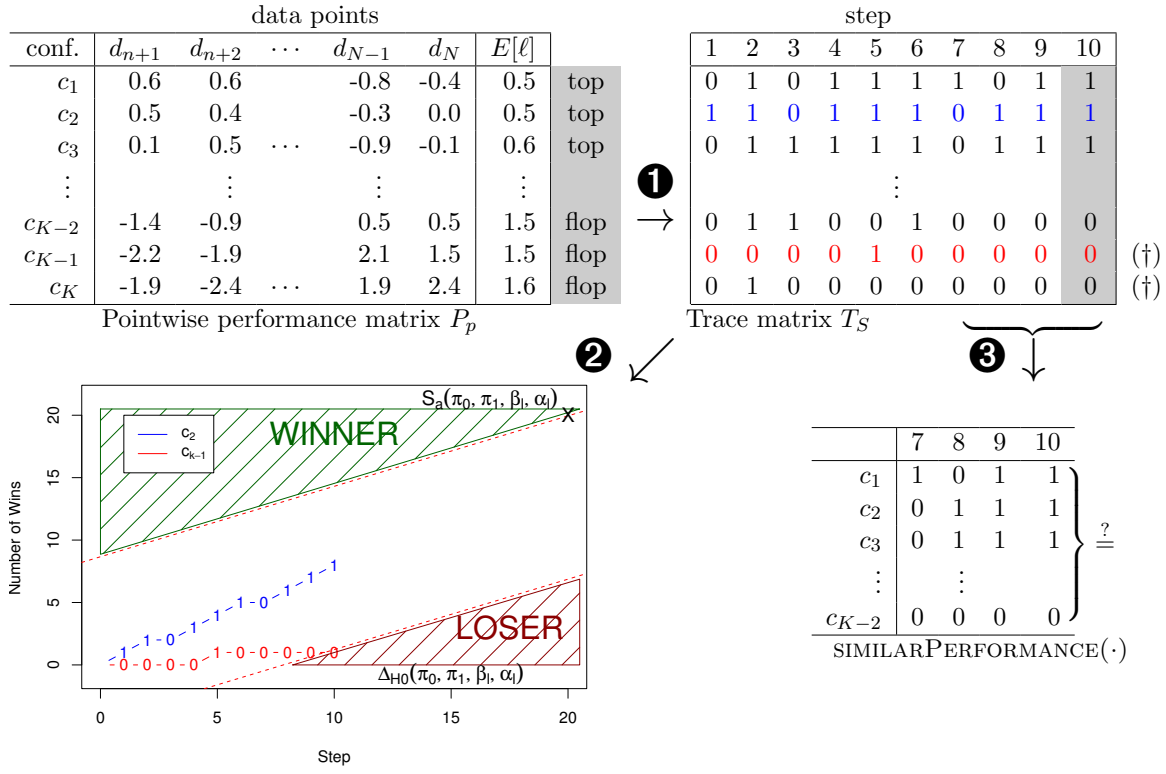


Figure 3: One step of CVST. Shown is the situation in step $s = 10$. ❶ A model based on the first n data points is learned for each configuration (c_1 to c_K). Test errors are calculated on the remaining data (d_{n+1} to d_N) and transformed into a binary performance indicator via robust testing. ❷ Traces of configurations are filtered via sequential analysis (c_{K-1} and c_K are dropped). ❸ The procedure checks whether the remaining configurations perform equally well in the past and stops if this is the case. See Appendix B for a complete example run.

see in the evaluation section that all these effects are indeed helpful for an overall good performance of the CVST algorithm.

To find the top performing configurations for step s we look at the outcome of the learned model for each configuration, i.e., we subsequently take the rows of the pointwise performance matrix P_p into account and apply either the Friedman test (Friedman, 1937) for regression experiments or the Cochran's Q test (Cochran, 1950) to see whether we observe statistically significant differences between configurations (see Appendix C for a summary of these tests). In essence these robust tests check whether the performance outcomes of a subset of the configurations show significant differences, i.e., in our case behave differently in terms of overall best performance. The assumption here is that the mean performance of a configuration is a good, yet wiggly estimator of its overall performance. By subsequently checking the finer-grained outcome of the models on the individual data points we want to find the breakpoint where the overall top-performing configurations for

this step are separated from the rest of the configurations which will show a significantly different behavior on the individual data points.

More formally, the function topConfigurations described in Algorithm 2 takes the pointwise performance matrix P_p as input and rearranges the rows according to the mean performances of the configurations yielding a matrix $\widetilde{P}_p$. Now for $k \in \{2, 3, \ldots, K\}$ we check, whether the first k configurations show a significantly different effect on the $N - n$ data points. This is done by executing either the Friedman test or the Cochran's Q test on the submatrix $\widetilde{P}_p[1:k, 1:(N-n)]$ with the pre-specified significance level α. If the test does not indicate a significant difference in the performance of the k configurations, we increment k by one and test again until we find a significant effect. Suppose we find a significant effect at index $\tilde{k}$. Since all previous tests indicated no significant effect for the $\tilde{k}-1$ configurations we argue that the addition of the $\tilde{k}^{\text{th}}$ configuration must have triggered the test procedure to indicate that in the set of these $\tilde{k}$ configurations is at least one configuration, which shows a significantly different behavior than all other configurations. Thus, we flag the configurations $1, \ldots, \tilde{k}-1$ as top configurations and the remaining $\tilde{k}, \ldots, K$ configurations as flop configurations. Note that this incremental procedure is a multiple testing situation, thus we apply the Bonferroni correction to the calculated p-values.

For the actual calculation of the test errors we apply an incremental model building process, i.e., the data added in each step on Line 22 increases the training data pool for each step by a set of size Δ. This would allow online algorithms to adapt their model also incrementally leading to even further speed improvements. The results of this first step are collected for each configuration in the trace matrix T_S (see Figure 3, top right), which shows the gradual transformation for the last 10 steps of the procedure highlighting the results of the last test. More formally, $T_S[c, s]$ is 1 iff configuration c is amongst the top configuration in step s; if c is not a top configuration in step s, the entry $T_S[c, s]$ is 0.

So this new column generated in step s in the trace matrix T_S summarizes the performance of all models learned on the first n data points in a robust way. Thus, the trace matrix T_S records the history of each configuration in a binary fashion, i.e., whether it performed as a top or flop configuration in each step of the CVST main loop. This leads to a robust transformation of the test errors of the configurations which can be modeled in the next step as a binary random variable with a success probability π indicating whether a configuration is amongst the top (high π) or the flop (low π) configurations.

4.2 Determining Significant Losers

Having transformed the test errors in a scale-independent top or flop scheme, we can now test whether a given parameter configuration is an overall loser. For this we represent a configuration as a binary random variable which turns out to be a top configuration with a given probability π. During the course of the execution of the CVST algorithm we gather information about the behavior of each configuration and want to estimate at each step whether the observed behavior is more likely to be associated with a binomial variable having a high π meaning that it is a winning configuration or a low one deeming it as a loser configuration. The standard tool for this kind of task is the sequential testing of binary random variables which is addressed in the *sequential analysis* framework developed by Wald (1947). Originally it has been applied in the context of production quality assessment

(compare two production processes) or biological settings (stop bioassays as soon as the gathered data leads to a significant result). In this section we focus on the general idea of this approach while Section 5 gives details about how the CVST algorithm deals with potential switches in the winning probability π of a given configuration.

The main idea of the sequential analysis framework is the following: One observes a sequence of i.i.d. Bernoulli variables $B_1, B_2, \ldots$, and wants to test whether these variables are distributed according to the hypotheses $H_0 : B_i \sim \pi_0$ or the alternative hypotheses $H_1 : B_i \sim \pi_1$ with $\pi_0 < \pi_1$ denoting the according success probabilities of the Bernoulli variables. Both significance levels for the acceptance of H_1 and H_0 can be controlled via the user-supplied meta-parameters α_l and β_l. The test computes the likelihood for the so far observed data and rejects one of the hypothesis when the respective likelihood ratio exceeds an interval controlled by the meta-parameters. It can be shown that the procedure has a very intuitive geometric representation, shown in Figure 3, lower left: The binary observations are recorded as cumulative sums at each time step. If this sum exceeds the upper red line L_1, we accept H_1; if the sum is below the lower red line L_0 we accept H_0; if the sum stays between the two red lines we have to draw another sample.

Wald's test requires that we fix both success probabilities π_0 and π_1 beforehand. Since our main goal is to use the sequential test to eliminate underperformers, we choose the parameters π_0 and π_1 of the test such that H_1 (a configuration wins) is postponed as long as possible. This will allow the CVST algorithm to keep configurations until the evidence of their performances definitely shows that they are overall loser configurations. At the same time, we want to maximize the area where configurations are eliminated (region $\triangle_{H_0}$ denoted by "LOSER" in Figure 3), rejecting as many loser configurations on the way as possible:

$$\begin{aligned}(\pi_0, \pi_1) \quad = \quad & \underset{\pi_0', \pi_1'}{\operatorname{argmax}} \ \triangle_{H_0}(\pi_0', \pi_1', \beta_l, \alpha_l) \\ & \text{s.t. } S_a(\pi_0', \pi_1', \beta_l, \alpha_l) \in (S-1, S]\end{aligned} \tag{1}$$

with $S_a(\cdot, \cdot, \cdot, \cdot)$ being the earliest step of acceptance of H_1 marked by an X in Figure 3 and S denotes again the total number of steps. By using approximations from Wald (1947) for the expected number of steps the test will take, if the real success probability of the underlying process would indicate a constant winner (i.e., $\pi = 1.0$), we can fix S_a to the maximal number of steps S and solve Equation (1) as follows (see Appendix D for details):

$$\pi_0 = 0.5 \wedge \pi_1 = \frac{1}{2}\sqrt[S]{\frac{1-\beta_l}{\alpha_l}}. \tag{2}$$

Equipped with these parameters for the sequential test, we can check each remaining trace on Line 18 of Algorithm 1 in the function isFlopConfiguration detailed in Algorithm 3 whether it is a statistically significant flop configuration (i.e., exceeds the lower decision boundary L_0) or not.

Note that sequential analysis formally requires i.i.d. variables. In the CVST procedure both the independence of the top/flop variable and the identically distributed assumption might be violated for configurations which transform to a winner configuration later on, thereby changing their behavior from a flop to a top configuration. With the modeling

approach taken in this step of the CVST algorithm this would amount to a change of the underlying success probability π of the configuration. Thus, the assumptions of the sequential testing framework would definitely be violated. We accommodate for this by introducing in Section 5.1 a so-called *safety zone* which acts as a safeguard against prematurely dropping of a configuration. Note that this safety zone can be controlled by the experimenter using the parameters α_l and β_l of the sequential test. If the experimenter chooses the right safety zone the underlying success probabilities of the configuration remain stable after the safety zone and, hence, again will satisfy the preconditions of the sequential testing framework. So by ensuring no premature drop of a configuration in the safety zone we heuristically adapt the sequential test to the potential switch of underlying success probabilities. To give a complete account of the assumptions of the sequential analysis we will discuss potential violations of the independence of the top/flop variables and its implication for the CVST procedure in Section 5.

For details of the open sequential analysis please consult Wald (1947) or see for instance Wetherill and Glazebrook (1986) for a general overview of sequential testing procedures. Appendix D contains the necessary details needed to implement the proposed testing scheme for the CVST algorithm.

4.3 Early Stopping and Final Winner

Finally, we employ an early stopping rule (Line 20) which takes the last w_{stop} columns from the trace matrix and checks whether all remaining configurations performed equally well in the past. In Figure 3 this submatrix of the overall trace matrix T_S is shown for a value of $w_{\text{stop}} = 4$ for the remaining configurations after step 10. For the test, we again apply the Cochran's Q test (see Appendix C) in the similarPerformance procedure on the submatrix of T_S as denoted in Algorithm 4. Figure 4 illustrates a complete run of the CVST algorithm for roughly 600 configurations. Each configuration marked in red corresponds to a flop configuration and a black one to a top configuration. Configurations marked in gray have been dropped via the sequential test during the CVST algorithm. The small zoom-ins in the lower part of the picture show the last w_{stop} remaining configurations during each step which are used in the evaluation of the early stopping criterion. We can see that the procedure keeps on going if there is a heterogeneous behavior of the remaining configurations (zoom-in is mixed red/black). When all the remaining configurations performed equally well in the past (zoom-in is nearly black), the early stopping test does not see a significant effect anymore and the procedure is stopped.

Finally, in the procedure selectWinner, Line 23 and Algorithm 5, the winning configuration is picked from the configurations which have survived all steps as follows: For each remaining configuration we determine the rank in a step according to the average performance during this step. Then we average the rank over the last w_{stop} steps and pick the configuration which has the lowest mean rank. This way, we make most use of the data accumulated during the course of the procedure. By restricting our view to the last w_{stop} observations we also take into account that the optimal parameter might change with increasing model size: Since we focus on the most recent observations with the biggest models, we always pick the configuration which is most suitable for the data size at hand.

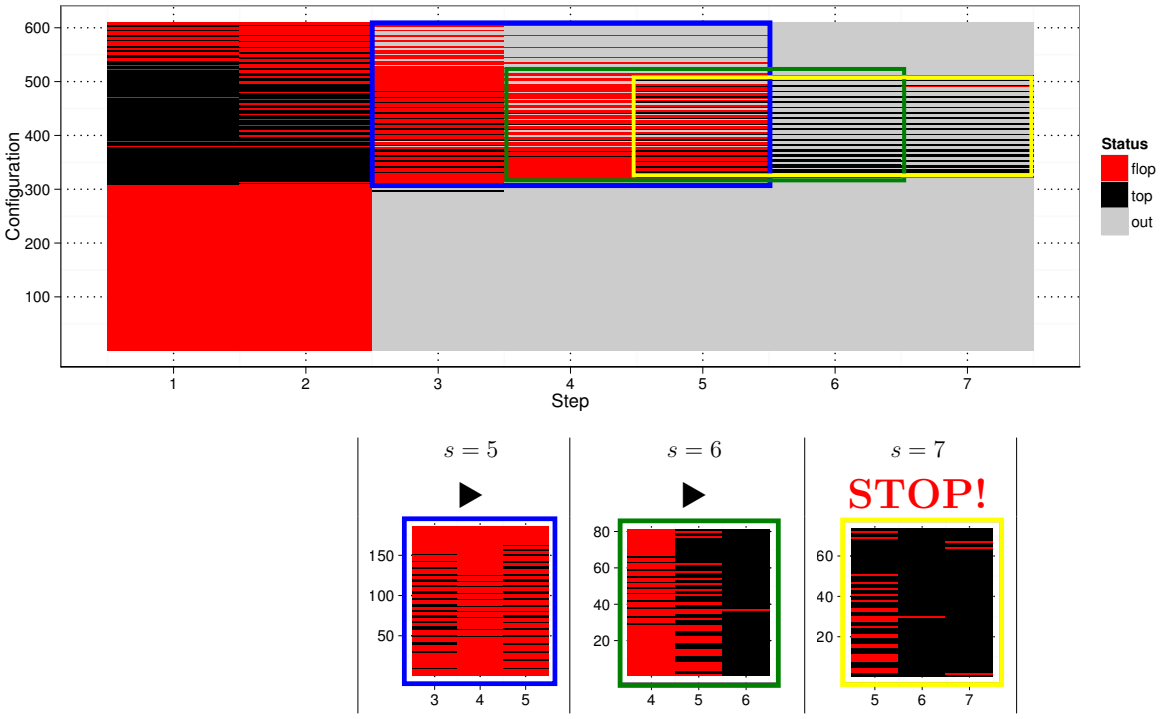


Figure 4: The upper plot shows a run of the CVST algorithm for roughly 600 configurations. At each step a configuration is marked as top (black), flop (red) or dropped (gray). The zoom-ins show the situation for step 5 to 7 without the dropped entries. The early stopping rule takes effect in step 7, because the remaining configurations performed equally well during step 5 to 7.

4.4 Meta-Parameters for the CVST

The CVST algorithm has a number of meta-parameters which the experimenter has to choose beforehand. In this section we give suggestions on how to choose these parameters. The parameter α controls the significance level for the test for similar behavior in each step of the procedure. We suggest to set this to the usual level of $\alpha = 0.05$. Furthermore β_l and α_l control the significance level of the H_0 (configuration is a loser) and H_1 (configuration is a winner) respectively. We suggest an asymmetric setup by setting $\beta_l = 0.1$, since we want to drop loser configurations relatively fast and $\alpha_l = 0.01$, since we want to be really sure when we accept a configuration as overall winner. Finally, we set w_{stop} to 3 for $S = 10$ and 6 for $S = 20$, as we have observed that this choice works well in practice.

5. Properties of the CVST Algorithm

After having introduced the overall concept of the CVST algorithm, we now focus on some properties of the procedure: Exploiting properties of the underlying sequential testing framework, we show how the experimenter can control the algorithm to work in a sta-

ble regime. Given some assumptions about the top/flop variables we show that the CVST algorithm performs with high accuracy after a configuration has reached its stable regime. Additionally, we show how the CVST algorithm can be used to work best on a given time budget. Finally, we discuss some unsolved questions and give possible directions for future research.

5.1 Performance in a Stable Regime

As discussed in Section 2 the winning probability of a configuration might change if we feed the learning algorithm more data. Therefore, a reasonable algorithm exploiting the learning on subsets of the data must be capable of dealing with these difficulties and potential change points in the behavior of certain configurations. In this section we investigate some properties of the CVST algorithm which makes it particularly suitable for learning on increasing subsets of the data.

The first property of the open sequential test employed in the CVST algorithm comes in handy to control the overall convergence process and to assure that no configurations are dropped prematurely:

Lemma 1 (Safety Zone) *Given the CVST algorithm with significance level α_l, β_l for being a top or flop configuration respectively, and maximal number of steps S, and a configuration which loses for the first s_{cp} iterations, as long as*

$$0 \leq \frac{s_{cp}}{S} \leq \frac{s_{safe}}{S} \text{ with } s_{safe} = \frac{\log \frac{\beta_l}{1-\alpha_l}}{\log 2 - \sqrt[S]{\frac{1-\beta_l}{\alpha_l}}} \text{ and } S \geq \left\lceil \log \frac{1-\beta_l}{\alpha_l} / \log 2 \right\rceil,$$

the probability that the configuration is dropped prematurely by the CVST algorithm is zero.

Proof The details of the proof are deferred to Appendix D. ■

The consequence of Lemma 1 is that the experimenter can directly control via the significance levels α_l, β_l until which iteration no premature dropping should occur and therefore guide the whole process into a stable regime in which the configurations will see enough data to show their real performance. Note that this property is a direct consequence of the sequential analysis framework and is used here to guide the test into a controlled region where we do not observe a premature dropping of configurations. Equation (2) ensures that we actually perform a meaningful test to discriminate a loser configuration ($\pi_0 = 0.5$) from a winning configuration ($\pi_1 > \pi_0$). Thus, by adjusting the safety zone of the CVST algorithm the experimenter can ensure that the configurations act according to the preconditions of the sequential testing framework introduced in Section 4.2, namely exhibiting a fixed probability π of being a winner configuration at each step.

Note that this safety zone is solely a guard for premature dropping of a configuration due to insufficient data in the first few steps of the CVST algorithm. The experimenter should have a notion at which point the performance of the configurations should stabilize in terms of error behavior, i.e., the learners see enough data to show their real performance. We argue that at this point the configurations behave reasonable stable, thus, fulfilling

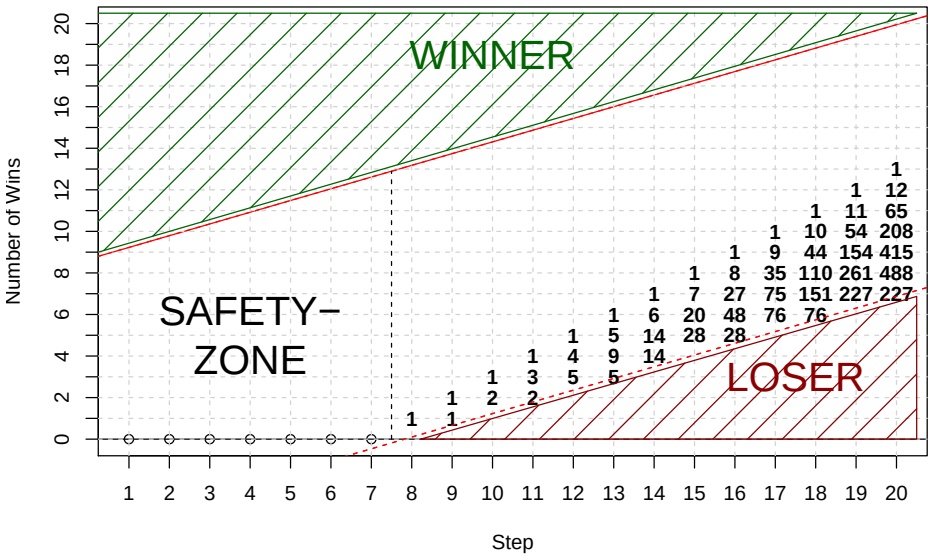


Figure 5: Visualization of the worst-case scenario for the error probability of the CVST algorithm: A global winner configuration is labeled as a constant loser until the safety zone is reached. Then we can calculate the probability that this configuration endures the sequential test by a recurrence scheme, which counts the number of remaining paths ending up in the non-loser region.

both the independence and identically distributed assumption of the sequential learning framework. We are fully aware that these assumptions are strong, yet, backed up by our extensive experimental evaluation in Section 6, we want to shed some light on why the CVST procedure shows such impressive speed-ups with small impact on the accuracy compared to ordinary cross-validation and even outperforms other model selection heuristics.

Hence, we define a *stable* configuration as a configuration which sticks to a certain probability π of being a winning configuration. So after having seen enough data to show its real behavior the robust transformation of the test error of the configuration inside the CVST algorithm (see Section 4.1) exhibits the properties of an i.i.d. Bernoulli variable and thus acts as a *stable* configuration in the subsequent steps of the CVST procedure. So a global winning configuration will be a *stable* configuration with a probability $\pi \gg \pi_0 = 0.5$.

Using these assumptions we can now take a look at the worst case performance of the CVST algorithm. Suppose a global winning configuration has been constantly marked as a loser up to the safety zone, because the amount of data available up to this point was not sufficient to show the superiority of this configuration. Given that the global winning configuration now sees enough data to be marked as a winning configuration by the binarization process throughout the next steps with probability π, we can give an error bound of the overall process by solving specific recurrences.

Figure 5 gives a visual impression of our worst case analysis for the example of a 20 step CVST execution: The winning configuration generated a straight line of zeros up to the

safety zone of 7. Our approach to bound the error of the fast cross-validation now consists essentially in calculating the probability mass that ends up in the non-loser region. The following lemma shows how we can express the number of paths which lead to a specific point on the graph by a two-dimensional recurrence relation:

Lemma 2 (Recurrence Relation) *Denote by* $\mathrm{Path}(s_R, s_C)$ *the number of paths, which lead to the point at the intersection of row* s_R *and column* s_C *and lie above the lower decision boundary* L_0 *of the sequential test. Given the worst case scenario described above the number of paths can be calculated as follows:*

$$\mathrm{Path}(s_R, s_C) = \begin{cases} 1 & \text{if } s_R = 0 \wedge c \leq s_{safe} = \frac{\log \frac{\beta_l}{1-\alpha_l}}{\log 2 - \sqrt[S]{\frac{1-\beta_l}{\alpha_l}}} \\ 1 & \text{if } s_R = s_C - s_{safe} \\ \mathrm{Path}(s_R, s_C - 1) + \mathrm{Path}(s_R - 1, s_C - 1) & \text{if } L_0(c) < s_R < s_C - s_{safe} \\ 0 & \text{otherwise.} \end{cases}$$

Proof We split the proof into the four cases:

1. The first case is by definition: The configuration has a straight line of zeros up to the safety zone s_{safe}.

2. The second case describes the diagonal path starting from the point $(1, s_{\mathrm{safe}} + 1)$: By construction of the paths (1 means diagonal up; 0 means one step to the right) the diagonal path can just be reached by a single combination, namely a straight line of ones.

3. The third case is the actual recurrence: If the given point is above the lower decision bound L_0, then the number of paths leading to this point is equal to the number of paths that lie directly to the left of this point plus the paths which lie directly diagonal downwards from this point. From the first paths this point can be reached by a direct step to the right and from the latter the current point can be reached by a diagonal step upwards. Since there are no other options than that by construction, this equality holds.

4. The last case describes all other paths, which either lie below the lower decision bound and therefore end up in the loser region or are above the diagonal and thus can never be reached.

■

This recurrence is visualized in Figure 5. Each number on the grid gives the number of valid, non-loser paths, which can reach the specific point. With this recurrence we are now able to prove a global, worst-case error probability of the fast cross-validation.

Theorem 3 (Error Bound of CVST for Stable Configuration) *Suppose a global winning configuration has reached the safety zone with a constant loser trace and then switches*

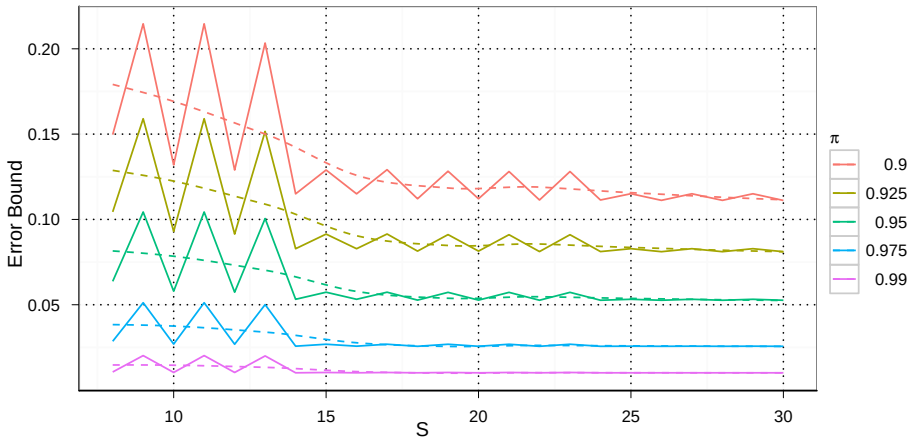


Figure 6: Error bound of the fast cross-validation as proven in Theorem 3 for different success probabilities π and maximal step sizes S. To mark the global trend we fitted a LOESS curve given as dotted line to the data.

to a stable winner configuration with a success probability of $\pi \gg \pi_0 = 0.5$. *Then the error that the CVST algorithm erroneously drops this configuration can be determined as follows:*

$$P(\textit{reject } \pi) \leq 1 - \sum_{i=\lfloor L_0(S) \rfloor + 1}^{r} \text{Path}(i, S)\pi^i(1-\pi)^{r-i} \qquad \textit{with } r = S - \left\lfloor \frac{\log \frac{\beta_l}{1-\alpha_l}}{\log 2 - \sqrt[S]{\frac{1-\beta_l}{\alpha_l}}} \right\rfloor .$$

Proof The basic idea is to use the number of paths leading to the non-loser region to calculate the probability that the configuration actually survives. This corresponds to the last column of the example in Figure 5. Since we model the outcome of the binarization process as a binomial variable with the success probability of π, the first diagonal path has a probability of π^r. The next paths each have a probability of $\pi^{(r-1)}(1-\pi)^1$ and so on until the last viable paths are reached in the point $(\lfloor L_0(S) \rfloor + 1, S)$. So the complete probability of the survival of the configuration is summed up with the corresponding number of paths from Lemma 2. Since we are interested in the complementary event, we subtract the resulting sum from one, which concludes the proof. ■

Note that the early stopping rule does not interfere with this bound: The worst case is indeed that the process goes on for the maximal number of steps S, since then the probability mass will be maximally spread due to the linear lower decision boundary and the corresponding exponents are maximal. So if the early stopping rule terminates the process before reaching the maximum number of steps, the resulting error probability will be lower than our given bound.

The error bound for different success probabilities and the proposed sequential test with $\alpha_l = 0.01$ and $\beta_l = 0.1$ are depicted in Figure 6. First of all we can observe a relatively fast convergence of the overall error with increasing maximal number of steps S. The impact on

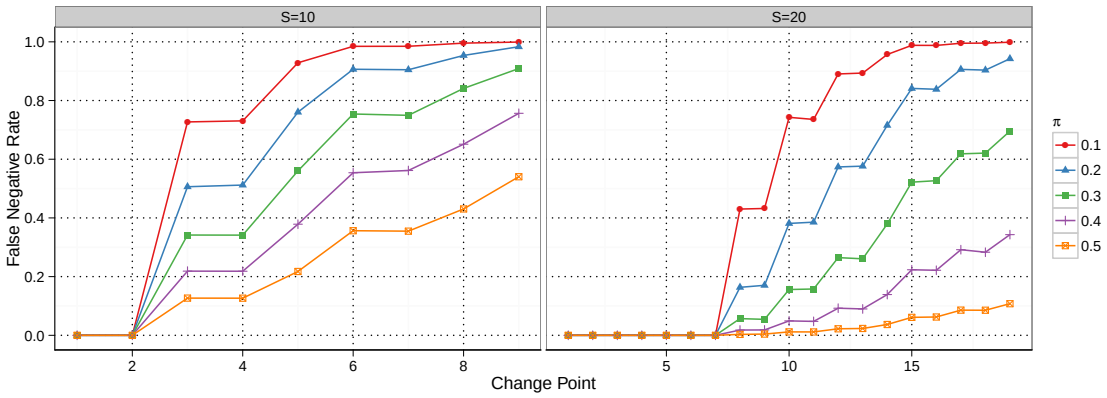


Figure 7: False negatives generated with the open sequential test for non-stationary configurations , i.e., at the given change point the Bernoulli variable changes its π_{before} from the indicated value to 1.0.

the error is marginal for the shown success probabilities, i.e., for instance for $\pi = 0.95$ the error nearly converges to the optimum of 0.05. Note that the oscillations especially for small step sizes originate from the rectangular grid imposed by the interplay of the Path-operator and the lower decision boundary L_0 leading to some fluctuations. Overall, the chosen test scheme allows us not only to control the safety zone but also has only a small impact on the error probability, which once again shows the practicality of the open sequential ratio test for the fast cross-validation procedure. By using this statistical test we can balance the need for a conservative retention of configurations as long as possible with the statistically controlled dropping of significant loser configurations with nearly no impact on the overall error probability.

Our analysis assumes that the experimenter has chosen the right safety zone for the learning problem at hand. For small data sizes it could happen that this safety zone was chosen too small, therefore the change point of the global winning configuration might lie outside the safety zone. While this will not occur often for today's sizes of data sets we have analyzed the behavior of CVST under this circumstances to give a complete view of the properties of the algorithm. To get insight into the drop rate for the case when the experimenter underestimated the change point s_{cp} we simulate those switching configurations by independent Bernoulli variables which change their success probability π from a chosen $\pi_{\text{before}} \in \{0.1, 0.2, \ldots, 0.5\}$ to a constant 1.0 at a given change point. This behavior essentially imitates the behavior of a switching configuration which starts out as a loser (i.e., up to the change point the trace will consist more or less of zeros) and after enough data is available turns into a constant winner.

The relative loss of these configurations for 10 and 20 steps is plotted in Figure 7 for different change points. The figure reveals our theoretical findings of Lemma 1 showing the corresponding *safety zone* for the specific parameter settings: For instance for $\alpha_l = 0.01$ and $\beta_l = 0.1$ and $S = 10$ steps, the safety zone amounts to 0.27×10, meaning that

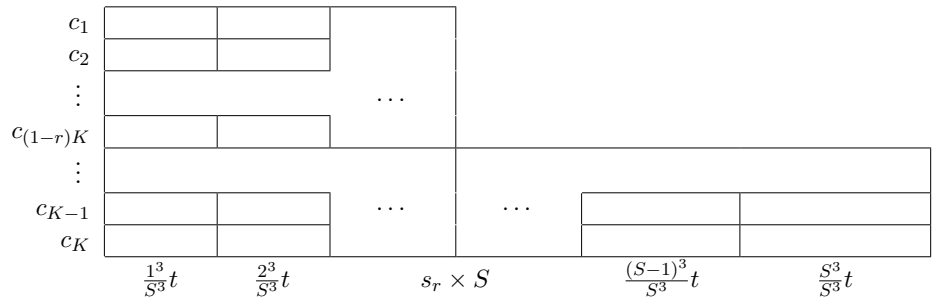


Figure 8: Approximation of the time consumption for a cubic learner. In each step we calculate a model on a subset of the data, so the model calculation time t on the full data set is adjusted accordingly. After $s_r \times S$ steps of the process, we assume a drop to $r \times K$ remaining configurations.

if the change point for all switching configurations occurs at step one or two, the CVST algorithm would not suffer from false positives. Similarly, for $S = 20$ the safety zone is $0.39 \times 20 = 7.8$. These theoretical results are confirmed in our simulation study, where the false negative rate is zero for sufficiently small change points for the open variant of the test. After that, there are increasing probabilities that the configuration will be removed. Depending on the success probability of the configuration before the change point, the resulting false negative rate ranges from mild for $\pi = 0.5$ to relatively severe for $\pi = 0.1$. The later the change point occurs, the higher the resulting false negative rate will be. Interestingly, if we increase the total number of steps from 10 to 20, the absolute values of the false negative rates are significantly lower. So even when the experimenter underestimates the actual change point, the CVST algorithm has some extra room which can even be extended by increasing the total number of steps.

5.2 Fast-Cross Validation on a Time Budget

While the CVST algorithm can be used out of the box to speed up regular cross-validation, the aforementioned properties of the procedure come in handy when we face a situation in which an optimal parameter configuration has to be found given a fixed computational budget. If the time is not sufficient to perform a full cross-validation or the amount of data that has to be processed is too big to explore a sufficiently spaced parameter grid with ordinary cross-validation in a reasonable time, the CVST algorithm can easily be adjusted to the specified time constraint. Thus, the experimenter is able to get the maximal number of model evaluations given the time budget available to judge which model is the best.

This is achieved by calculating a maximal steps parameter S which leads to a near coverage of the available time budget T as depicted in Figure 8. The idea is to specify an expected drop rate $(1 - r)$ of configurations and a safety zone bound s_{safe}. Then we can give a rough estimate of the total time needed for a CVST with a total number of steps S, equating this with the available time budget T and solving for S. More formally, given K parameter configurations and a pre-specified safety zone bound $s_{\text{safe}} = s_r \times S$ with

$0 < s_r < 1$ to ensure that no configuration is dropped prematurely, the computational demands of the CVST algorithm are approximated by the sum of the time needed before step s_{safe} involving the model calculation of all K configurations and after step s_{safe} for $r \times K$ configurations with $0 < r < 1$. As we will see in the experimental evaluation section, this assumption of a given drop rate of $(1 - r)$ leading to the form of time consumption as depicted in Figure 8 is quite common. The observed drop rate corresponds to the overall difficulty of the problem at hand.

Given the computation time t needed to perform the model calculation on the full data set, we prove in Appendix E that the optimal maximum step parameter for a learner of time complexity $f(n) = n^m$ can be calculated as follows:

$$S = \left\lfloor \frac{m+1}{4} \frac{2T - tK(1-r)s_r^m + tKr}{((1-r)s_r^{m+1} + r)tK} + \sqrt{\left[\frac{m+1}{4} \frac{2T - tK(1-r)s_r^m + tKr}{((1-r)s_r^{m+1} + r)tK}\right]^2 - \frac{m(m+1)}{12} \frac{(1-r)s_r^{m-1} + r}{(1-r)s_r^{m+1} + r}} \right\rfloor .$$

After calculating the maximal number of steps S given the time budget T, we can use the results of Lemma 1 to determine the maximal β_l given a fixed α_l, which yields the requested safety zone bound s_{safe}.

5.3 Discussion of Further Theoretical Analyses

In Section 2 we have noted that in order for the test performances to converge, the parameter configurations should be independent of the sample size. As shown in Appendix A this holds for a range of standard methods in machine learning. Yet, special care has to be taken to really ensure this assumption. For instance for kernel ridge regression one has to scale the ridge parameter during each step of the CVST algorithm to accommodate for the change in the learning set size (see the reference implementation in the official CRAN package named `CVST` or the development version at `https://github.com/tammok/CVST`). The ν-Support Vector Machine on the other hand directly incorporates this scaling of parameters which makes it a good fit for the CVST algorithm. Generally, it would be preferable to have this scaling automatically incorporated in the CVST algorithm such that the experimenter could plug-in his favorite method without the need to think about any scaling issues of hyper-parameters. Unfortunately, this is highly algorithm dependent and, thus, is an open problem for further research.

An additional concern to the practitioner is how to choose the correct size of the safety zone s_{safe}. If the training set does not contain enough data to get to a stable regime of the parameter configurations, even regular cross-validation on the full data set would yield incorrect configurations. But if we have just barely enough data to reach this stable region, setting the right safety zone is essential for the CVST algorithm to return the correct configurations. Unfortunately we are not aware of any test or bound which could hint at the right safety zone given a data set and learner. Yet, in today's world of big data where sample sizes are more often too big than too small, this might not pose a serious problem anymore. Nevertheless, we have analyzed the behavior of the CVST algorithm in case the experimenter underestimates the safety zone in Section 5.1 showing that even for these cases CVST is able to absorb a certain amount of misspecification.

The similarity test introduced in Section 4.1 relies on two assumptions: First, the averaged loss function over the data not used for training in one step gives us a good indicator of the performance of a configuration. Second, well performing configurations show similar behavior in classification or regression on the data not used for learning. While these assumptions definitely make sense, they encode a certain optimism of how the grid of configurations is populated: If we have too few configurations as input to the procedure it might happen that some non-optimal configurations mask out the other, normally optimal, configurations just by chance. To overcome this problem we therefore would need a certain amount of redundancy in the configuration grid. Both the amount of redundancy and thus the similarity measure underlying this redundancy assumption are hard to grasp theoretically, yet, it could lead to new ways to model the binary transformation of the performance of configurations in each step of the CVST algorithm.

There might be even further potential in the behavior of similar configurations that could be used in the CVST algorithm: If there is a notion of similarity between different configurations, it would be interesting to exploit this information and incorporate it into the CVST algorithm. For instance, one could add this kind of information in the function topConfigurations of Algorithm 1 to average the result of similar configurations and, hence, extend the pooling effect of the test already available for the data point dimension in the direction of configurations.

While the selection scheme explained in Section 4.2 deals with the fact of potential change points of a configuration, it is not clear how independent the individual entries of a trace for a given configuration are and how much these potential dependencies influence the power of the sequential testing framework. Preliminary experiments comparing the CVST algorithm as described in this paper and a version of the CVST algorithm where at each step the data pool is shuffled, thus, yielding always different data points for learning and evaluation, showed no significant differences between these two versions. This indicates that at least the potential dependencies introduced by the overlap of learning sets due to subsequent addition of data points do not interfere with the dependency assumption of the sequential testing framework. We will see in the evaluation section that the CVST procedure in its current form shows excellent behavior throughout a wide range of data sets; yet, further research of the theoretical properties of CVST might yield even better procedures in the future.

6. Experiments

Before we evaluate the CVST algorithm on real data, we investigate its performance on controlled data sets. Both for regression and classification tasks we introduce special tailored data sets to highlight the overall behavior and to stress-test the fast cross-validation procedure. To evaluate how the choice of learning method influences the performance of the CVST algorithm, we compare kernel logistic regression (KLR) against a ν-Support Vector Machine (SVM) for classification problems and kernel ridge regression (KRR) versus ν-SVR for regression problems each using a Gaussian kernel (see Roth, 2001; Schölkopf et al., 2000). In all experiments we use a 10 step CVST with parameter settings as described in Section 4.4 (i. e. $\alpha = 0.05, \alpha_l = 0.01, \beta_l = 0.1, w_{\text{stop}} = 3$) to give us an upper bound of the expected speed gain. Note that we could get even higher speed gains by either lowering the

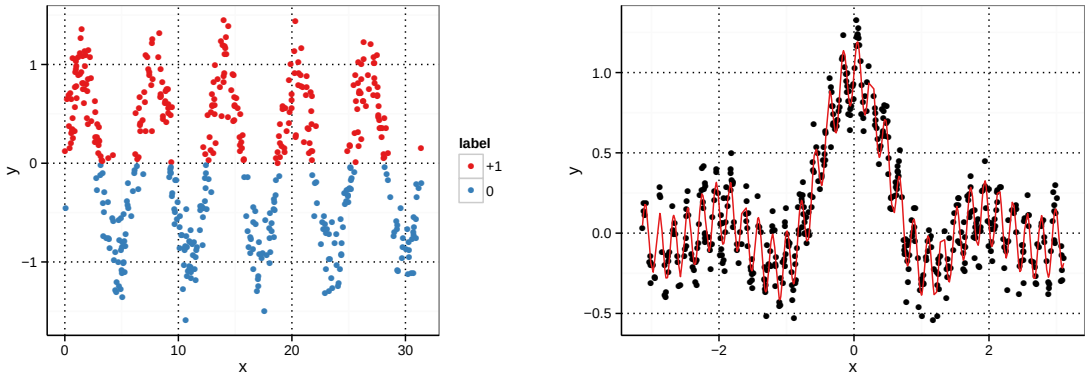


Figure 9: The *noisy sine* (left) and *noisy sinc* data set (right).

number of steps or increasing β_l. From a practical point of view we believe that the settings studied are highly realistic.

6.1 Artificial Data Sets

To assess the quality of the CVST algorithm we first examine its behavior in a controlled setting. We have seen in our motivation section that a specific learning problem might have several layers of structure which can only be revealed by the learner if enough data is available. For instance in Figure 2(a) we can see that the first optimal plateau occurs at $\sigma = 0.1$, while the real optimal parameter centers around $\sigma = 0.01$. Thus, the real optimal choice just becomes apparent if we have seen more than 200 data points.

In this section we construct a learning problem both for regression and classification tasks which could pose severe problems for the CVST algorithm: If it stops too early, it will return a suboptimal parameter set. We evaluate how different intrinsic dimensionalities of the data and various noise levels affect the performance of the procedure. For classification tasks we use the *noisy sine* data set, which consists of a sine uniformly sampled from a range controlled by the intrinsic dimensionality d:

$$y = \sin(x) + \epsilon \quad \text{with } \epsilon \sim \mathcal{N}(0, n^2), x \in [0, 2\pi d], n \in \{0.25, 0.5\}, d \in \{5, 50, 100\}.$$

The labels of the sampled points are just the sign of y. An example for $d = 5, n = 0.25$ is plotted in the left subplot of Figure 9. For regression tasks we devise the *noisy sinc* data set, which consists of a sinc function overlayed with a high-frequency sine:

$$y = \operatorname{sinc}(4x) + \frac{\sin(15dx)}{5} + \epsilon \text{ with } \epsilon \sim \mathcal{N}(0, n^2), x \in [-\pi, \pi], n \in \{0.1, 0.2\}, d \in \{2, 3, 4\}.$$

An example for $d = 2, n = 0.1$ is plotted in the right subplot of Figure 9. For each of these data sets we generate 1,000 data points and run a 10 step CVST and compare its results with a normal 10-fold cross-validation on the full data set. We record both the test error on additional 10,000 data points and the time consumed for the parameter search. The explored parameter grid contains 610 equally spaced parameter configurations

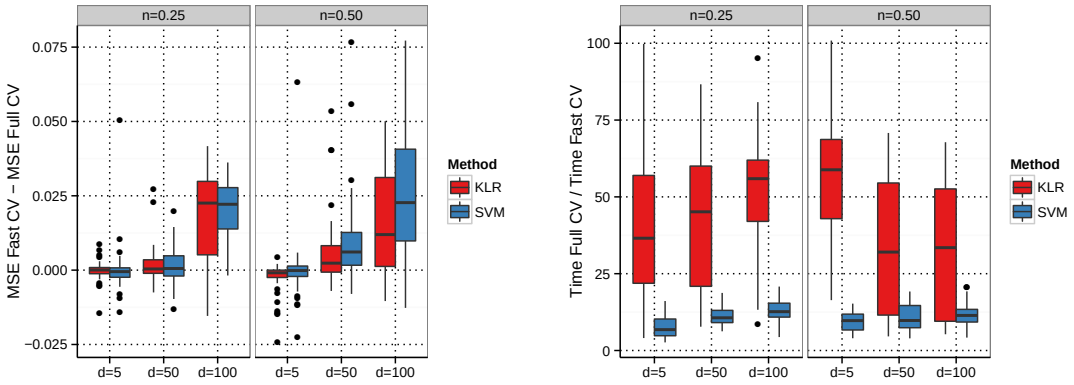


Figure 10: Difference in mean square error (left) and relative speed gain (right) for the *noisy sine* data set.

for each method ($\log_{10}(\sigma) \in \{-3, -2.9, \ldots, 3\}$ and $\nu \in \{0.05, 0.1, \ldots, 0.5\}$ for SVM/SVR and $\log_{10}(\lambda) \in \{-7, -6, \ldots 2\}$ for KLR/KRR, respectively). This process is repeated 50 times to gather sufficient data for an interpretation of the overall process. Apart from recording the difference in mean square error (MSE) of the learner selected by normal cross-validation and by the CVST algorithm we also look at the relative speed gain. Note that we have encoded the classes as 0 and 1 for the classification experiments so the MSE corresponds to the misclassification rate of the learner. So the difference in MSE gives us a good measurement of the impact of using the CVST algorithm for both classification and regression experiments.

The results for the *noisy sine* data set can be seen in Figure 10. The left boxplots show the distribution of the difference in MSE of the best parameter determined by CVST and normal cross-validation. In the low noise setting ($n = 0.25$) the CVST algorithm finds the same optimal parameter as the normal cross-validation up to the intrinsic dimensionality of $d = 50$. For $d = 100$ the CVST algorithm gets stuck in a suboptimal parameter configuration yielding an increased classification error compared to the normal cross-validation. This tendency is slightly increased in the high noise setting ($n = 0.5$) yielding a broader distribution. The classification method used seems to have no direct influence on the difference, both SVM and KLR show nearly similar behavior. This picture changes when we look at the speed gains: While the SVM nearly always ranges between 10 and 20, the KLR shows a speed-up between 20 and 70 times. The variance of the speed gain is generally higher compared to the SVM which seems to be a direct consequence of the inner workings of KLR: The main loop performs at each step a matrix inversion of the whole kernel matrix until the calculated coefficients converge. Obviously this convergence criterion leads to a relative wide-spread distribution of the speed gain when compared to the SVM performance.

Figure 11 shows the distribution of the number of remaining configurations after each step of the CVST algorithm. In the low noise setting (upper row) we can observe a tendency of higher drop rates up to $d = 100$. For the high noise setting (lower row) we observe a

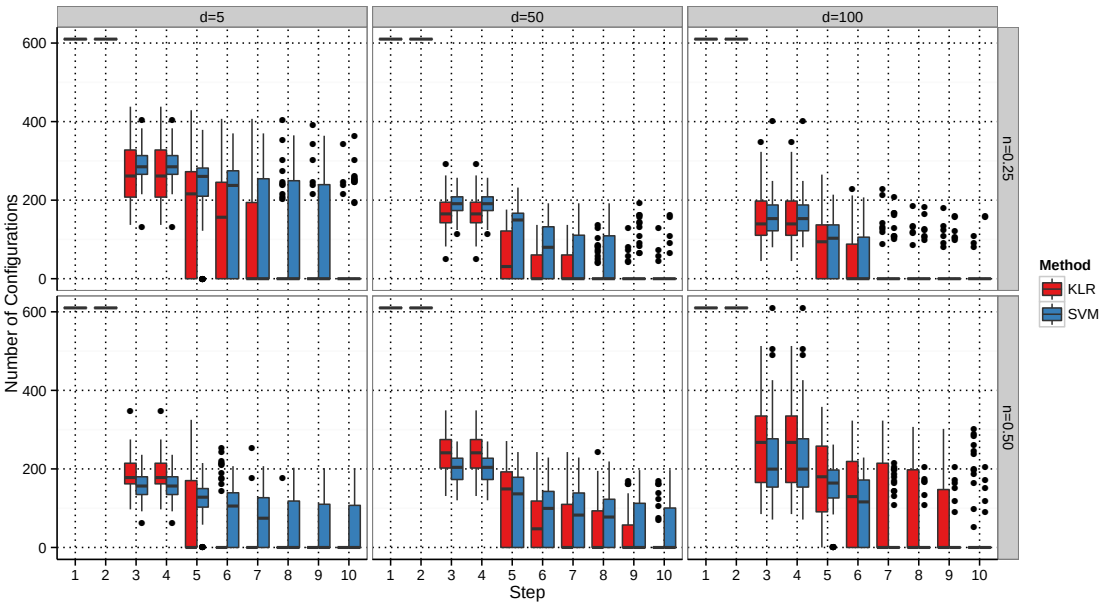


Figure 11: Remaining configurations after each step for the *noisy sine* data set.

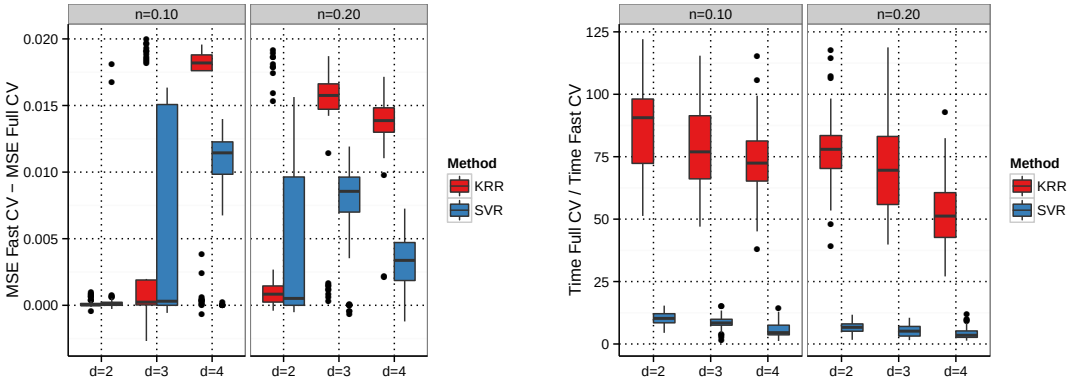


Figure 12: Difference in mean square error (left plots) and relative speed gain (right plots) for the *noisy sinc* data set.

steady increase of kept configurations combined with a higher spread of the distribution. Overall we see a very effective drop rate of configurations for all settings. The SVM and the KLR show nearly similar behavior so that the higher speed gain of the KLR we have seen before is a consequence of the algorithm itself and is not influenced by the CVST algorithm.

The performance on the *noisy sinc* data set is shown in Figure 12. The first striking observation is the transition of the CVST algorithm which can be observed for the intrinsic

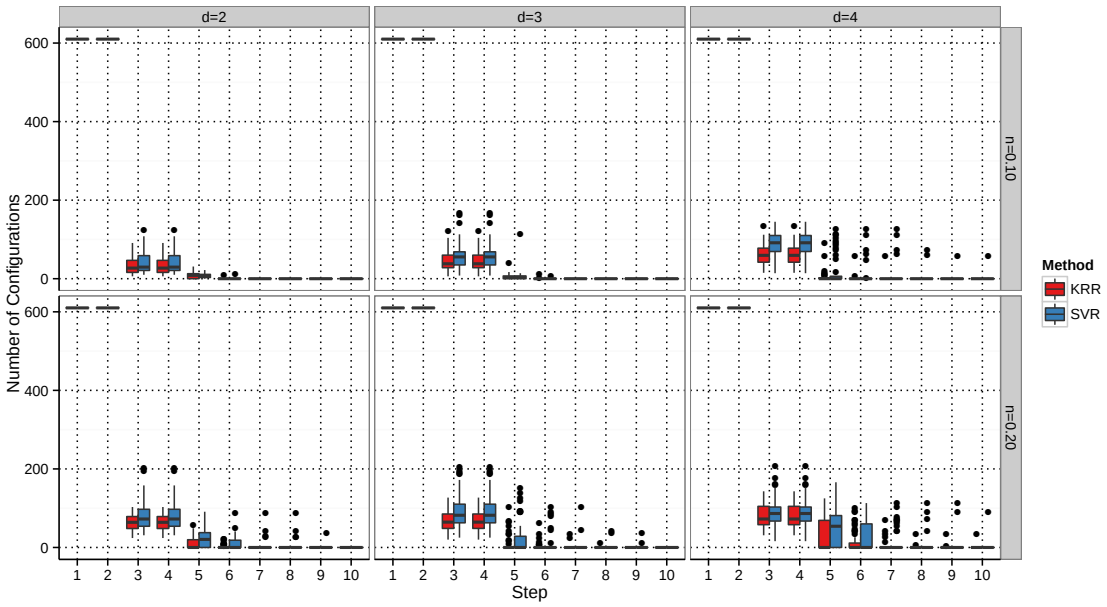


Figure 13: Remaining configurations after each step for the *noisy sinc* data set.

dimensionality of $d = 3$. At this point the overall excellent performance of the CVST algorithm is on the verge of choosing a suboptimal parameter configuration. This behavior is more evident in the high noise setting. In the case of SVR the difference to the solution found by the normal cross-validation is always smaller than for KRR. The speed gain observed shows a small decline over the different dimensionalities and noise levels and ranges between 10 and 20 for the SVR and 50 to 100 for KRR.

This is a direct consequence of the behavior which can be observed in the number of remaining configurations shown in Figure 13. Compared to the classification experiments the drop is much more drastic. The intrinsic dimensionality and the noise level show a small influence (higher dimensionality or noise level yields more remaining configurations) but the overall variance of the distribution is much smaller than in the classification experiments.

In Figure 14 we examine the influence of more data on the performance of the CVST algorithm. Both for the *noisy sine* and *noisy sinc* data set we are able to estimate the correct parameter configuration for all noise and dimensionality settings if we feed the CVST with enough data.[1] Clearly, the CVST is capable of extracting the right parameter configuration if we increase the amount of data to 2000 or 5000 data points, rendering our method even more suitable for big data scenarios: If data is abundant, CVST will be able to estimate the correct parameter in a much smaller time frame.

1. Note that we have to limit this experiment to the SVM/SVR method, since the full cross-validation of the KLR/KRR would have taken too much time to compute.

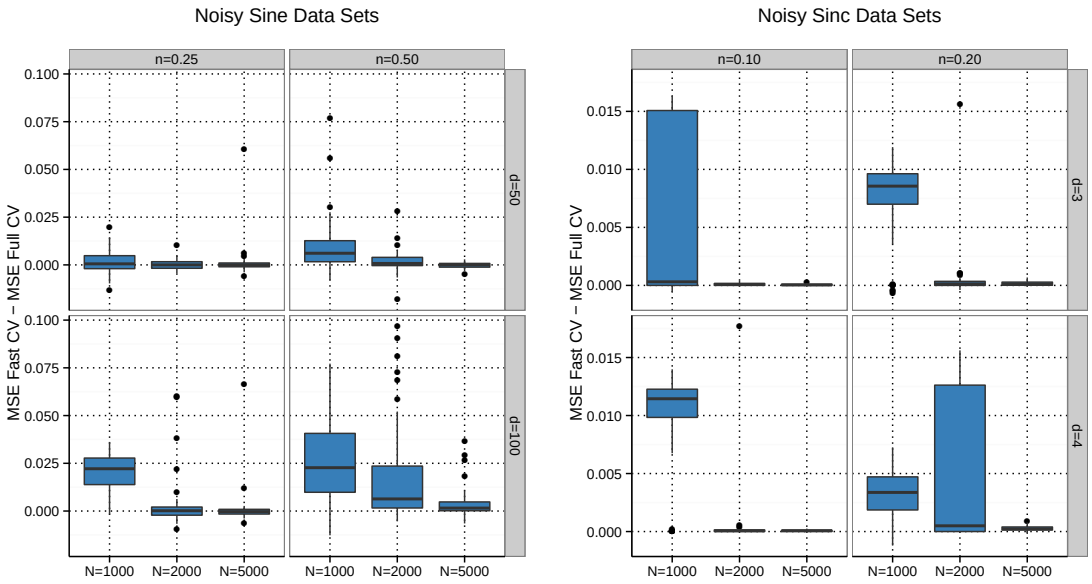


Figure 14: Difference in mean square error for SVM/SVR with increasing data set size for *noisy sine* (left) and the *noisy sinc* (right) data sets. By adding more data, the CVST algorithm converges to the correct parameter configuration.

6.2 Benchmark Data Sets

After demonstrating the overall performance of the CVST algorithm on controlled data sets we will investigate its performance on real life and well known benchmark data sets. For classification we picked a representative choice of data sets from the IDA benchmark repository (see Rätsch et al. 2001[2]). Furthermore we added the first two classes with the most entries of the *covertype* data set (see Blackard and Dean, 1999). Then we follow the procedure of the paper in sampling 2,000 data points of each class for the model learning and estimate the test error on the remaining data points. For regression we pick the data used in Donoho and Johnstone (1994) and add the *bank32nm*, *pumadyn32nm* and *kin32nm* of the Delve repository.[3]

We process each data set as follows: First we normalize each variable of the data to zero mean and variance of one, and in case of regression we also normalize the dependent variable. Then we split the data set in half and use one part for training and the other for the estimation of the test error. This process is repeated 50 times to get sufficient statistics for the performance of the methods. As in the artificial data setting we compare the speed gain of the fast compared to the normal cross-validation on the same parameter grid of 610 values. To allow for better comparability of the performance on the different data sets we report the mean square error (MSE) ratio of the CVST procedure compared to the normal

2. Available at `http://www.mldata.org`.
3. Available at `http://www.cs.toronto.edu/~delve`.

Data	Method	MSE Ratio	Speed	Data	Method	MSE Ratio	Speed
banana	KLR	$\mathbf{0.998 \pm 0.012}$	61.1	bank	KRR	0.994 ± 0.003	85.7
banana	SVM	$\mathbf{1.001 \pm 0.008}$	16.9	bank	SVR	0.998 ± 0.001	43.9
covtype	KLR	$\mathbf{0.994 \pm 0.011}$	84.3	blocks	KRR	0.762 ± 0.022	81.3
covtype	SVM	0.939 ± 0.009	53.4	blocks	SVR	0.886 ± 0.016	39.5
german	KLR	$\mathbf{0.987 \pm 0.017}$	27.5	bumps	KRR	0.784 ± 0.043	86.1
german	SVM	$\mathbf{0.981 \pm 0.024}$	5.5	bumps	SVR	0.666 ± 0.030	37.3
image	KLR	$\mathbf{1.032 \pm 0.051}$	33.5	doppler	KRR	0.766 ± 0.035	92.4
image	SVM	0.923 ± 0.060	13.8	doppler	SVR	0.937 ± 0.014	41.2
ringnorm	KLR	0.814 ± 0.050	111.7	heavisine	KRR	0.981 ± 0.005	53.2
ringnorm	SVM	$\mathbf{0.999 \pm 0.017}$	32.4	heavisine	SVR	0.988 ± 0.003	33.7
splice	KLR	$\mathbf{0.991 \pm 0.019}$	52.2	kin	KRR	0.994 ± 0.002	58.7
splice	SVM	$\mathbf{1.005 \pm 0.016}$	18.1	kin	SVR	0.996 ± 0.001	39.9
twonorm	KLR	$\mathbf{1.017 \pm 0.034}$	50.1	pumadyn	KRR	0.992 ± 0.003	68.2
twonorm	SVM	$\mathbf{1.015 \pm 0.014}$	25.7	pumadyn	SVR	0.984 ± 0.007	29.6
waveform	KLR	$\mathbf{0.989 \pm 0.014}$	54.0				
waveform	SVM	$\mathbf{0.992 \pm 0.013}$	22.8				

Table 2: Comparison of performance of the CVST algorithm to full cross-validation (classification data sets in left part, regression data sets in right part). MSE ratio is the relative gain in MSE of CVST compared to the full cross-validation. Speed denotes the relative speed increase of CVST compared to the full cross-validation. We report the mean values over 50 repetitions and 1.96 standard errors. If CVST performs on par or better than the full cross-validation (i.e., MSE ratio plus 1.96 standard errors is bigger than 1.0) the values are in boldface.

cross-validation, i.e., values over 1.0 favor the CVST procedure. For the *blocks*, *bumps*, and *doppler* data set of Donoho and Johnstone (1994) we adjusted the range of σ to a smaller scale ($\log_{10}(\sigma) \in \{-6, -5.9, \ldots, 0\}$) to have reasonable results in the parameter grid of 610 values since these data sets contain a very fine-grained structure. Note that this adjustment is just for the sake of comparability to the other data sets.

Figure 15 shows the result for the classification data sets (left side) and the regression data sets (right side). The upper panels depict the relative gain in MSE of CVST compared to the full cross-validation. For the classification tasks we see that CVST is on par with the full cross-validation except for the SVM for *covtype* and KLR for *ringnorm*. For the regression task we observe that except for the *blocks*, *bumps* and *doppler* data sets CVST chooses reasonable parameter set. Although for some problems the CVST algorithm picks a suboptimal parameter set, even then the relative performance decreases are always relatively small and range around 80%. The learners have hardly any impact on the behavior; just for the *ringnorm* and the *blocks*, *bumps* and *doppler* data set we see a strong difference of the corresponding methods. These findings can also be observed in Table 2: Given the mean of the relative MSE ratio and the speed ratio with its corresponding 1.96 standard errors we mark entries in boldface where the MSE ratio plus the 1.96 standard error is bigger than 1.0 indicating a performance on par or better than the normal cross-validation. While CVST can tackle most of the classification task we see a relative decline in the regression tasks.

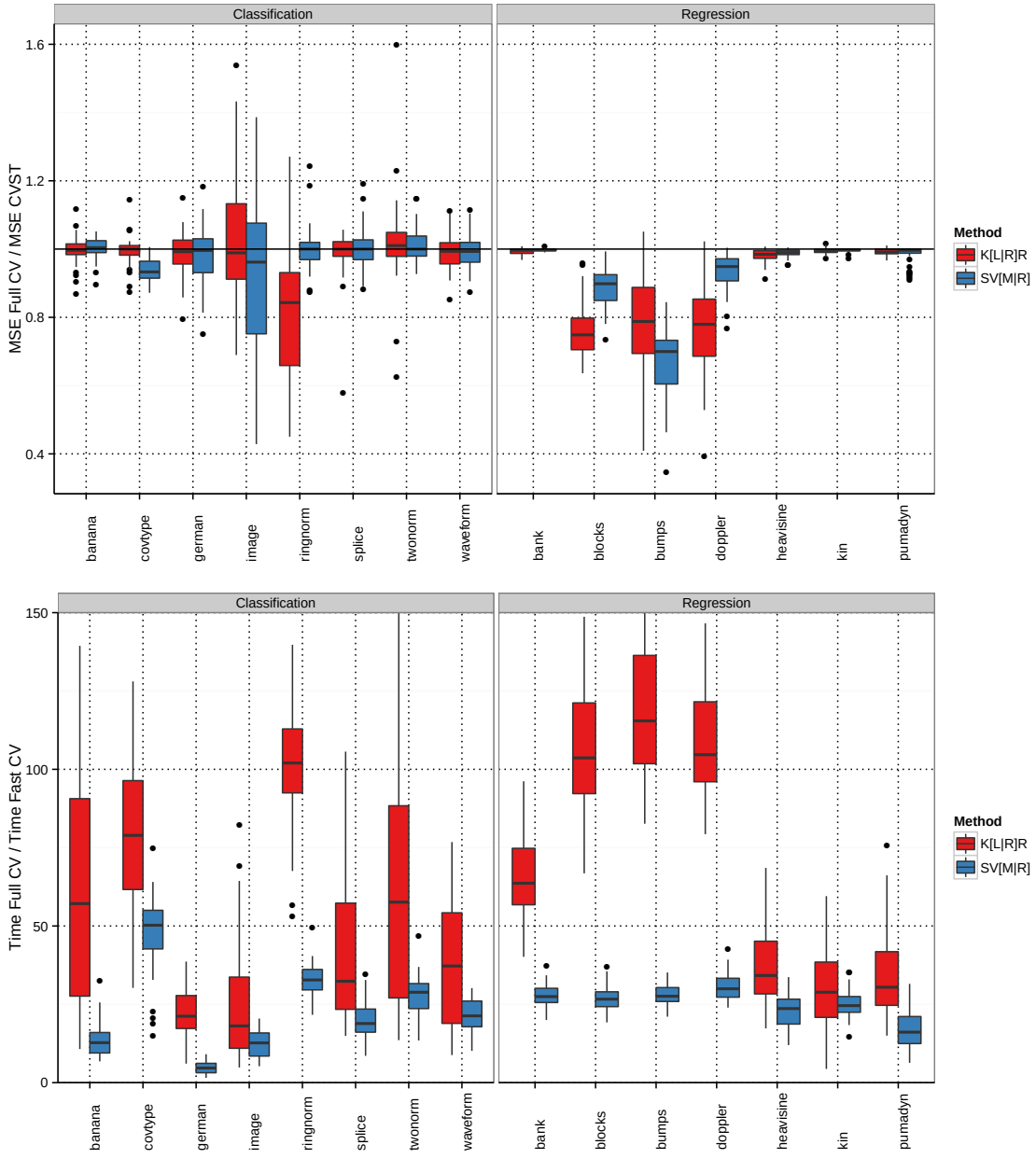


Figure 15: MSE ratio (upper plots) and relative speed gain (lower plots) for the *benchmark* data sets.

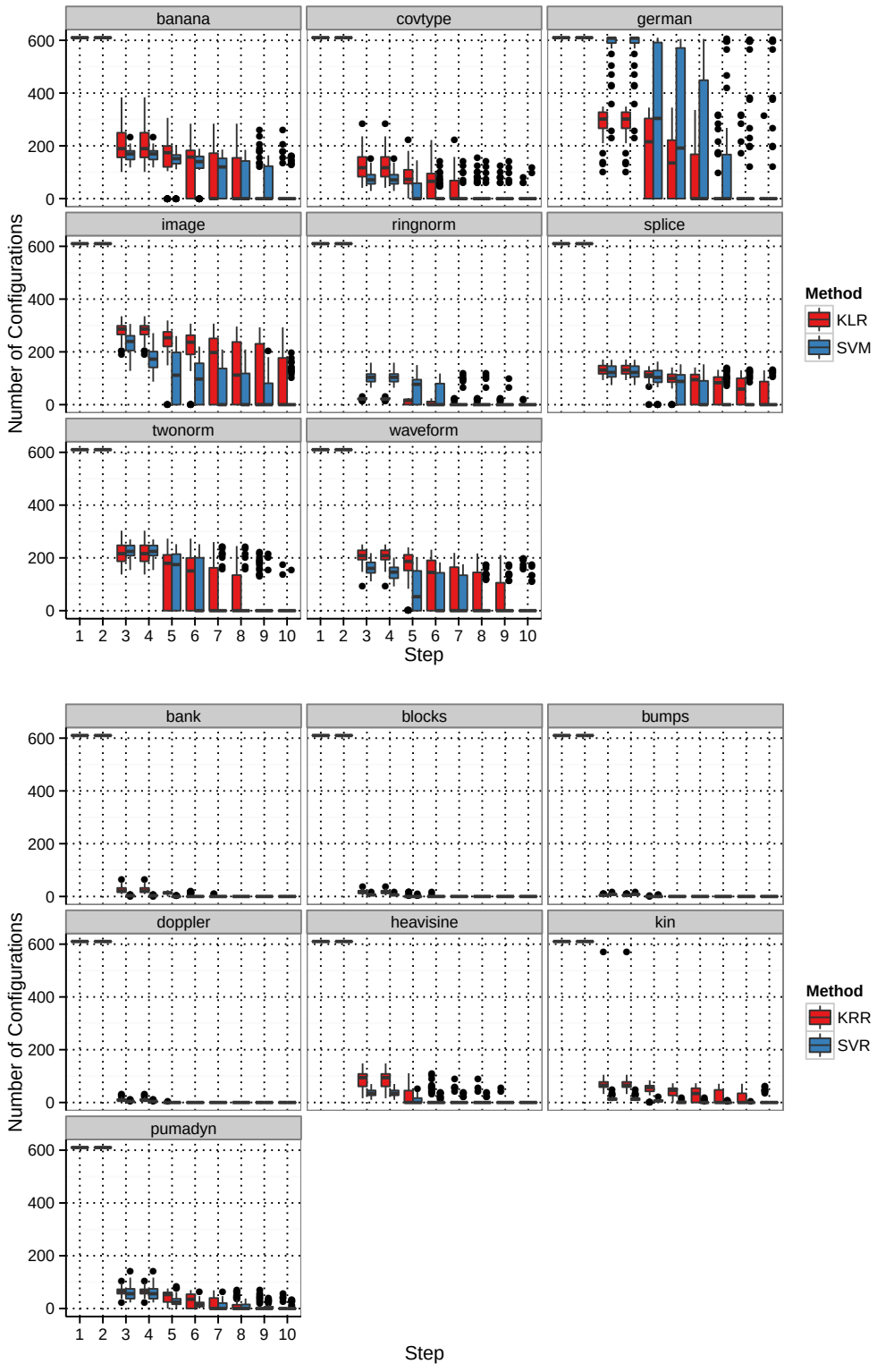


Figure 16: Remaining configurations after each step for different *benchmark* data sets.

But except for the *blocks*, *bumps* and *doppler* data sets the relative performance decrease ranges around 99% indicating a nearly optimal performance.

In terms of speed gain we see a much more diverse and varying picture. Overall, the speed improvements for KLR and KRR are higher than for SVM and SVR and reach up to 120 times compared to normal cross-validation. Regression tasks in general seem to be solved faster than classification tasks, which can clearly be explained when we look at the traces in Figure 16: For classification tasks the number of kept configurations is generally much higher than for the regression tasks. Furthermore we can observe several types of difficulty of the learning problems. For instance the *german* data set seems to be much more difficult than the *ringnorm* data (see Braun et al., 2008) which is also reflected in the difference and speed improvement seen in the previous figure. We will see in Section 7.1 that we can trade time for increasing the accuracy of CVST for the regression tasks by leveraging the modular construction of the CVST procedure.

Since finding the top configurations inside the loop of the CVST algorithm is a crucial step to the overall performance of the procedure we further investigate how our choice of tests influence the performance of the CVST procedure. Recall from Algorithm 2 that we used the Cochran's Q test for classification experiments (see line 9) and Friedman test for regression problems (see line 11) to find the top configurations in an iterative testing scheme. Both these test are non-parametric and paired tests, i.e., they both take into account the pointwise performance of a configuration and, thus, compare the performance on individual data points. In Section 4.1 we argued that this pooling effect robustifies the estimation of the top configurations since outliers in the testing data do not have such a dramatic effect on the test results compared to using for instance the overall test error as input for the test. To verify this claim we have replaced the tests in the Algorithm 2 by unpaired, non-parametric versions which solely test whether the test error is significantly different without taking the results of the individual data points into account. To this end we have replaced the Cochran's Q test for classification experiments on line 9 of Algorithm 2 by an unpaired version described by Wilson (1927) and the Friedman test by the Kruskal-Wallis rank sum test (Kruskal and Wallis, 1952). Again we repeat the procedure for each benchmark set 50 times to get reliable statistics.

The results are reported in Table 3, upper part. We can see that the relative level of MSE is almost always around 1.0 if we take the 1.96 standard error ranges into account. The upper plot of Figure 17 shows the distribution of the MSE ratio. Except for the *bumps* and *doppler* data set all distributions are clearly centered around 1.0 with a narrow spread which further highlights the equality of the two methods in terms of accuracy.

Comparing the ratio of the CVST procedure to the unpaired variant we can see that the paired test variant improves the runtime of the procedure significantly. This clearly demonstrates that the usage of the paired tests which directly estimate the top configurations on the pointwise predictions saves computation time with no impact on the accuracy.

Since we compared the CVST algorithm to a full cross-validation it is also of interest to see how CVST compares to a simple heuristic which uses just 10% of the data for the cross-validation. We have executed this experiment for all benchmark data sets and repeated the procedure 50 times to get statistically sound estimates. The middle part of Table 3 reports the MSE ratio of the CVST compared to the 10% cross-validation annotated with their corresponding 1.96 standard errors and again the speed ratio. The first striking thing to

observe is that the MSE ratios are all significantly bigger than 1.0, indicating that CVST always finds a better performing configuration than the simple 10% heuristic. While the accuracy impact varies across the different data sets CVST always picks significantly better performing configurations with a modest impact on the runtime compared to the simple 10% heuristic. This trend can clearly be seen in the middle plot of Figure 17 which shows the corresponding distributions of the relative level of MSE. For all data sets the bulk of the distribution is above 1.0 indicating better performance of the CVST method compared to the 10% heuristic.

The last comparison of the performance of the CVST method is shown in the lower part of Table 3. Here we show the MSE ratio of CVST compared to a random search as described in Bergstra and Bengio (2012). In each step of the random search procedure we choose parameters uniformly distributed over the range of the corresponding grid of the CVST procedure and learn a full-data model. After having spent the same amount of time as the CVST procedure on a specific data set we stop the random search. Then we pick the best model of the so far evaluated parameters and compare its performance to the CVST model. Again, the lower part of Table 3 shows the relative gain in MSE and their corresponding 1.96 standard errors gathered over 50 repetitions for each data set. Both the table and the lower part of Figure 17 indicate that CVST shows better performance compared to random search especially in the regression data sets. In some cases (*covtype* with SVM, *ringnorm* with SVM and *splice* with SVM) the random search outperforms CVST but in the majority of cases the CVST procedure can extract better parameter configurations in the same amount of time than the random search.

In summary, the evaluation of the benchmark data sets shows that the CVST algorithm gives a huge speed improvement compared to the normal cross-validation. While we see some non-optimal choices of configurations, the total impact on the error is never exceptionally high. We have to keep in mind that we have chosen the parameters of our CVST algorithm to give an impression of the maximal attainable speed-up: More conservative settings would trade computational time for lowering the impact on the test error. The CVST outperforms both unpaired variants of the procedure, the simple heuristics of cross-validation on just 10% of the data, and a random search in parameter space. This clearly demonstrates that the individual parts of the CVST procedure are well chosen and the combination of tests are superior to other methods. Trading some speed compared to simpler heuristics for more robust and stable estimates of optimal performing configurations and the huge speed improvement compared to a full cross-validation renders the CVST procedure as a promising candidate for model selection in big data settings.

7. Modularization and Extensions

In this Section we will deal with several aspects of the CVST algorithm: We illuminate the inner structure of the overall procedure and discuss potential extensions and properties of specific steps. The CVST algorithm consists of a sequence of tightly coupled modules: The output of the top or flop test is the input for the subsequent test for significant losers. The performance history of all remaining configurations is then the input for the early stopping rule which looks for similar performance of the remaining configurations on the learning problem to capture the right point in time to stop the CVST loop. This stepwise procedure

Data	Method	MSE Ratio	Speed	Data	Method	MSE Ratio	Speed
banana	KLR	1.001 ± 0.010	2.9	bank	KRR	0.984 ± 0.005	3.0
banana	SVM	1.005 ± 0.009	1.6	bank	SVR	1.000 ± 0.001	1.2
covtype	KLR	1.003 ± 0.005	1.6	blocks	KRR	0.957 ± 0.030	1.4
covtype	SVM	0.992 ± 0.007	1.2	blocks	SVR	0.986 ± 0.006	1.4
german	KLR	0.993 ± 0.013	0.9	bumps	KRR	0.920 ± 0.074	1.0
german	SVM	0.984 ± 0.019	0.6	bumps	SVR	0.999 ± 0.003	1.1
image	KLR	1.011 ± 0.040	3.3	doppler	KRR	0.954 ± 0.051	1.3
image	SVM	1.030 ± 0.047	1.5	doppler	SVR	0.981 ± 0.010	1.1
ringnorm	KLR	0.963 ± 0.037	1.1	heavisine	KRR	0.990 ± 0.005	2.6
ringnorm	SVM	1.005 ± 0.013	1.0	heavisine	SVR	0.995 ± 0.003	6.9
splice	KLR	0.992 ± 0.020	1.6	kin	KRR	1.000 ± 0.003	6.8
splice	SVM	1.012 ± 0.014	1.1	kin	SVR	1.000 ± 0.000	3.8
twonorm	KLR	0.988 ± 0.029	0.9	pumadyn	KRR	1.003 ± 0.003	26.0
twonorm	SVM	1.001 ± 0.010	0.9	pumadyn	SVR	1.000 ± 0.001	18.4
waveform	KLR	0.997 ± 0.013	1.5	*Unpaired Version of CVST*			
waveform	SVM	1.011 ± 0.012	1.3				
banana	KLR	$\mathbf{1.056 \pm 0.036}$	0.7	bank	KRR	$\mathbf{1.073 \pm 0.024}$	0.6
banana	SVM	$\mathbf{1.106 \pm 0.086}$	0.3	bank	SVR	$\mathbf{1.006 \pm 0.003}$	0.7
covtype	KLR	$\mathbf{1.062 \pm 0.028}$	0.9	blocks	KRR	$\mathbf{1.428 \pm 0.102}$	0.7
covtype	SVM	$\mathbf{1.031 \pm 0.020}$	0.4	blocks	SVR	$\mathbf{1.189 \pm 0.033}$	0.7
german	KLR	$\mathbf{1.085 \pm 0.031}$	1.5	bumps	KRR	$\mathbf{1.947 \pm 0.380}$	0.7
german	SVM	$\mathbf{1.132 \pm 0.061}$	0.6	bumps	SVR	$\mathbf{1.049 \pm 0.015}$	0.7
image	KLR	$\mathbf{1.544 \pm 0.185}$	0.4	doppler	KRR	$\mathbf{1.896 \pm 0.186}$	0.7
image	SVM	$\mathbf{1.279 \pm 0.143}$	0.8	doppler	SVR	$\mathbf{1.210 \pm 0.035}$	0.7
ringnorm	KLR	$\mathbf{2.083 \pm 0.291}$	1.3	heavisine	KRR	$\mathbf{1.104 \pm 0.022}$	0.4
ringnorm	SVM	$\mathbf{1.044 \pm 0.038}$	0.5	heavisine	SVR	$\mathbf{1.042 \pm 0.011}$	0.5
splice	KLR	$\mathbf{1.106 \pm 0.067}$	0.6	kin	KRR	$\mathbf{1.074 \pm 0.030}$	0.3
splice	SVM	$\mathbf{1.092 \pm 0.039}$	0.6	kin	SVR	$\mathbf{1.014 \pm 0.006}$	0.7
twonorm	KLR	$\mathbf{1.197 \pm 0.103}$	0.7	pumadyn	KRR	$\mathbf{1.053 \pm 0.016}$	0.4
twonorm	SVM	$\mathbf{1.032 \pm 0.026}$	0.4	pumadyn	SVR	$\mathbf{1.026 \pm 0.007}$	0.4
waveform	KLR	$\mathbf{1.099 \pm 0.036}$	0.7	*Cross-Validation on 10% of Data*			
waveform	SVM	$\mathbf{1.088 \pm 0.063}$	0.4				
banana	KLR	$\mathbf{1.198 \pm 0.131}$	1.0	bank	KRR	$\mathbf{1.339 \pm 0.068}$	1.0
banana	SVM	0.970 ± 0.012	1.0	bank	SVR	$\mathbf{3.070 \pm 0.376}$	1.0
covtype	KLR	$\mathbf{1.185 \pm 0.044}$	1.0	blocks	KRR	$\mathbf{2.059 \pm 0.529}$	1.0
covtype	SVM	0.886 ± 0.032	1.0	blocks	SVR	$\mathbf{1.526 \pm 0.314}$	1.0
german	KLR	$\mathbf{1.128 \pm 0.098}$	1.0	bumps	KRR	$\mathbf{2.567 \pm 0.629}$	1.0
german	SVM	1.020 ± 0.024	1.0	bumps	SVR	$\mathbf{4.461 \pm 0.511}$	1.0
image	KLR	$\mathbf{2.188 \pm 0.512}$	1.0	doppler	KRR	$\mathbf{1.891 \pm 0.405}$	1.0
image	SVM	$\mathbf{1.133 \pm 0.064}$	1.0	doppler	SVR	$\mathbf{2.337 \pm 0.583}$	1.0
ringnorm	KLR	$\mathbf{5.972 \pm 2.315}$	1.0	heavisine	KRR	$\mathbf{1.075 \pm 0.012}$	1.0
ringnorm	SVM	0.648 ± 0.094	1.0	heavisine	SVR	1.012 ± 0.043	1.0
splice	KLR	$\mathbf{2.062 \pm 0.389}$	1.0	kin	KRR	$\mathbf{1.283 \pm 0.049}$	1.0
splice	SVM	0.874 ± 0.022	1.0	kin	SVR	$\mathbf{1.190 \pm 0.033}$	1.0
twonorm	KLR	2.427 ± 1.444	1.0	pumadyn	KRR	$\mathbf{1.113 \pm 0.026}$	1.0
twonorm	SVM	0.939 ± 0.027	1.0	pumadyn	SVR	$\mathbf{1.070 \pm 0.025}$	1.0
waveform	KLR	$\mathbf{2.032 \pm 0.600}$	1.0	*Random Search*			
waveform	SVM	0.959 ± 0.019	1.0				

Table 3: Comparison of performance of the CVST algorithm to different competitors (details see text). MSE ratio is the relative gain in MSE of CVST compared to the other variant. Speed denotes the relative speed increase of CVST compared to the other variant. We report the mean values over 50 repetitions and 1.96 standard errors with significant better values of CVST in boldface.

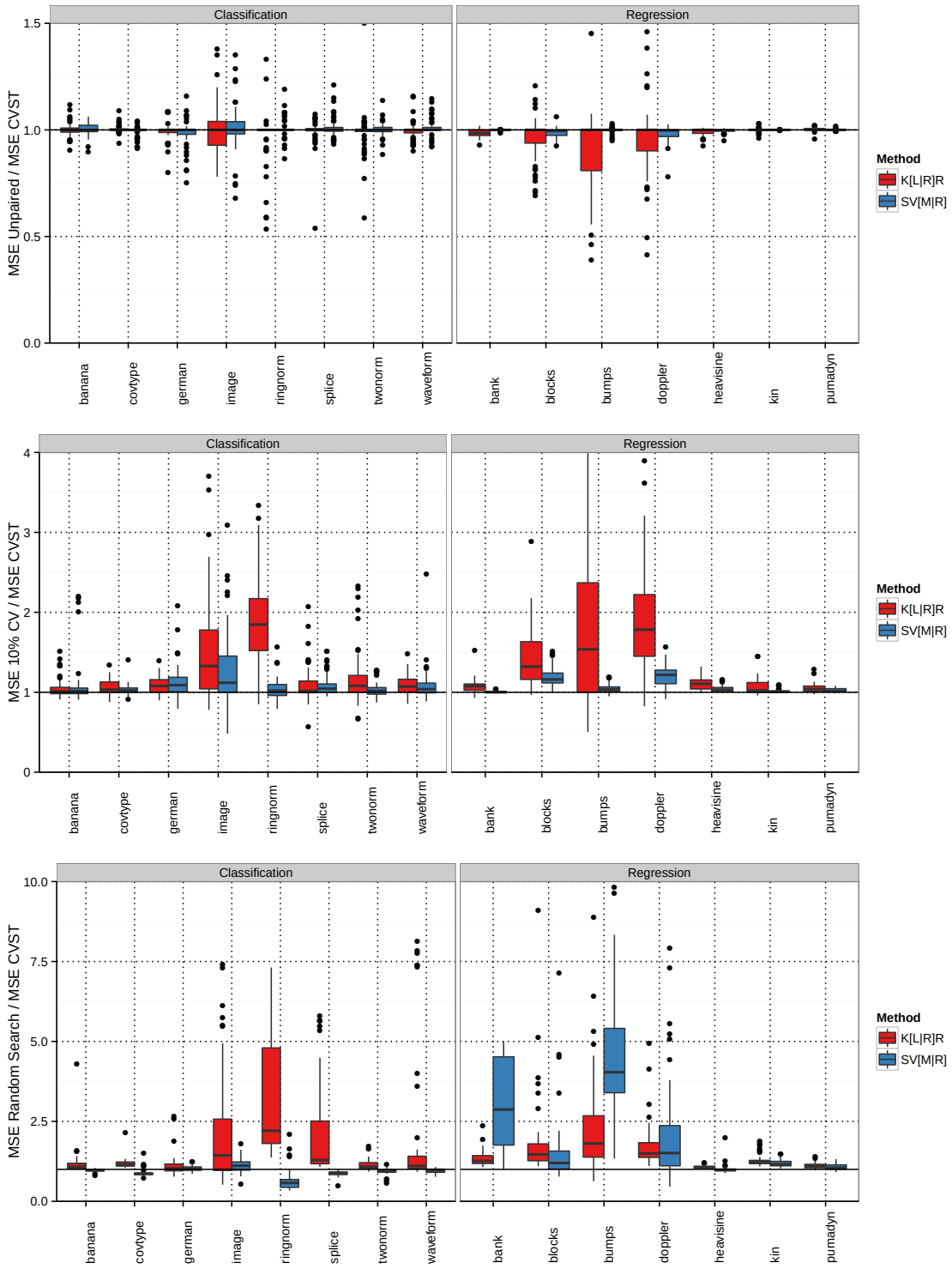


Figure 17: Distributions of the MSE ratio of the CVST procedure compared to the unpaired variant (upper panel), the cross-validation on 10% of the data (middle panel) and random search (lower panel). The horizontal line denotes the 1.0, i.e., equal performance ratio. Values over 1.0 favor the CVST procedure.

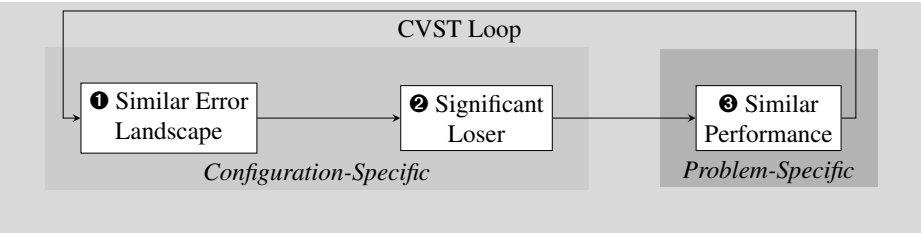


Figure 18: Conceptual view of the CVST algorithm. Each execution of the loop body consists of a sequence of test, each delivering the input for the following test. This modular structure allows for customization of the CVST algorithm to special situations (multi-class experiments, structured learning etc.).

is depicted in Figure 18: While the tests for top or flop configurations (step ❶) and the following sequential analysis (step ❷) focuses solely on the individual configurations, the early stopping rule (step ❸) acts on a global scope by determining the right point to stop the CVST algorithm. Thus, we face two kinds of test, namely the configuration-specific and the problem-specific tests.

To complete our discussion of the CVST algorithm, we focus on the configuration-specific procedures. First, we analyze the inner structure of the similarity test based on the error landscape in Section 7.1 and how this module can be adjusted for specific side constraints. Furthermore, in Section 7.2 we look at the suitability of the sequential analysis for determining significant loser configurations. It is shown that a so-called *closed* sequential test lacks essential properties of the open variant of Wald used in the CVST algorithm, which further underlines the appropriateness of the open test of Wald for the learning on increasing subsets of data.

7.1 Checking the Similarity of the Error Landscape

In the evaluation of the CVST method in Section 6 we see that the Friedman test for the regression case shows a much more aggressive behavior than the Cochran's Q test used in the top or flop conversion in the classification case. This feature can be clearly seen in Figure 16 where the dropping rates of the classification and regression benchmark data sets can be easily compared. Since the Friedman test acts on the squared residuals it uses more information compared to the classification task where we just have the information whether a specific data point was correctly classified or not. Thus, the Friedman test can exploit the higher detail of the information and can decide much faster than the Cochran's Q test which of the configurations are significantly different from the top performing ones.

In this section we show how the modular design of the CVST algorithm can be utilized to fit a less aggressive, yet more robust similarity test for regression data into the overall framework. It comes as no surprise that this increased tolerance affects the runtime of the CVST procedure. In the following we will first develop the alternative similarity test and then compare its performance both on the toy and the benchmark data sets to the original Friedman variant.

Recall from Section 4.1 that the top or flop assignment was calculated in a sequential manner: First we order all remaining configurations according to their mean performance; then we check at which point the addition of another configuration shows a significantly different behavior compared to all other, better performing configurations. To employ a less strict version of the Friedman test we drop the actual residual information and instead use the outlier behavior of a configuration for comparison. To this end we assume that the residuals are normally distributed with mean zero and a configuration-dependent variance σ_c^2 which we estimate from the actual residuals. Now we can check for each calculated residual whether it exceeds the $\frac{\alpha}{2}$ confidence interval around zero by using the normality assumption, thus converting the raw residuals in a binary information whether it is deemed as an outlier or not. Similar to the classification case this binary matrix forms the input to the Cochran's Q test which then asserts whether a specific configuration belongs to the top-performing ones or not.

The results of this procedure on the *noisy sinc* data set is shown in Figure 19: Compared to the outcome of the Friedman test in Figure 12 we can clearly see that the conservative nature of the outlier-based test helps in finding the correct parameter configuration. Obviously its higher retention rate leads to lower runtime performance: The speed ratio drops roughly by a factor of $\frac{2}{3}$. A similar behavior can be observed on the *benchmark* data sets in Figure 20: The conservative behavior of the outlier-based measure increases the MSE ratio compared to the residual-based test, but at the same time lowers the speed ratio. Interestingly, for the *benchmark* data sets the speed impact on the SVR is much lower compared to the speed ratio decrease of the KRR method. We can observe this shift also in the number of kept configurations shown in Figure 21 both for the *noisy sinc* and the *benchmark* data sets.

The conclusion of this discussion is two-fold: First, this section shows how the modular construction of the CVST methods allows for the exchange of the individual parts of the algorithm without disrupting the workflow of the procedure. If the residual-based test turns out to be unsuitable for a given regression problem, it is extremely easy to devise an adapted version for instance by looking at the outlier behavior of the configurations. Second, we see the inherent flexibility of the CVST algorithm. If there is a need for different error measures (for instance multi-class experiments, structured learning etc.), the modularized structure of the CVST algorithms allows for maximal flexibility and adaptability to special cases.

7.2 Determining Significant Losers: Open versus Closed Sequential Testing

As already introduced in Section 4.2 the sequential testing was pioneered by Wald (1947); the test monitors a likelihood ratio of a sequence of i.i.d. Bernoulli variables $B_1, B_2, \ldots$:

$$\ell = \prod_{i=1}^{n} f(b_i, \pi_1) / \prod_{i=1}^{n} f(b_i, \pi_0) \quad \text{given } H_h : B_i \sim \pi_h, h \in \{0, 1\}.$$

Hypothesis H_1 is accepted if $\ell \geq A$ and contrary H_0 is accepted if $\ell \leq B$. If neither of these conditions apply, the procedure cannot accept either of the two hypotheses and needs more data. A and B are chosen such that the error probability of the two decisions does not exceed α_l and β_l respectively. In Wald and Wolfowitz (1948) it is proven that the open sequential probability ratio test of Wald is optimal in the sense that compared to all tests

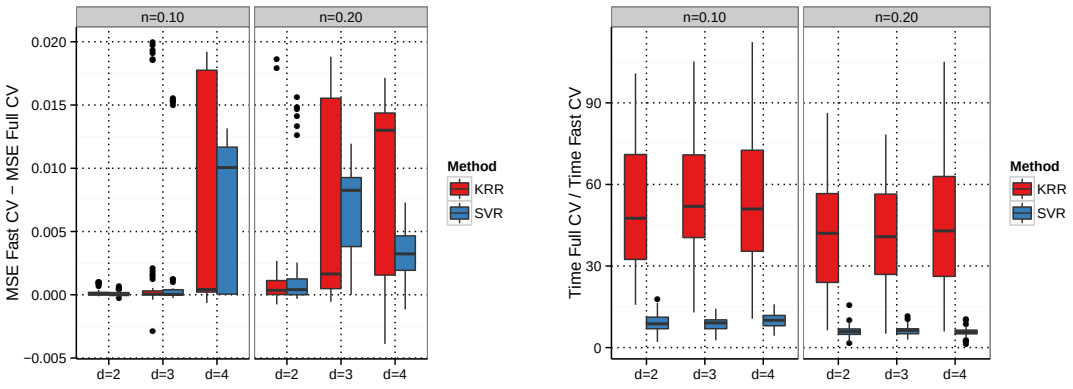


Figure 19: Difference in mean square error (left plots) and relative speed gain (right plots) for the *noisy sinc* data set using the outlier-based similarity test. In comparison to the stricter Friedman test used in Figure 12 we can observer a more conservative behavior resulting in increased robustness at the expense of performance.

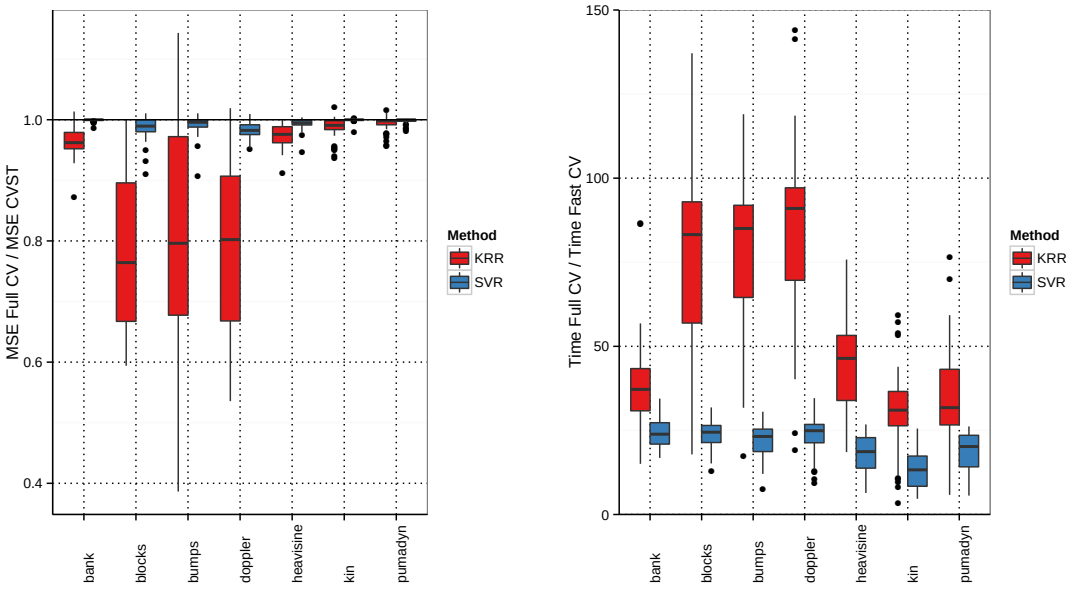


Figure 20: MSE ratio (left plot) and relative speed gain (right plot) for the *benchmark* data sets using the outlier-based similarity test. Compared to Figure 15 we can see better accuracy but decreased speed performance.

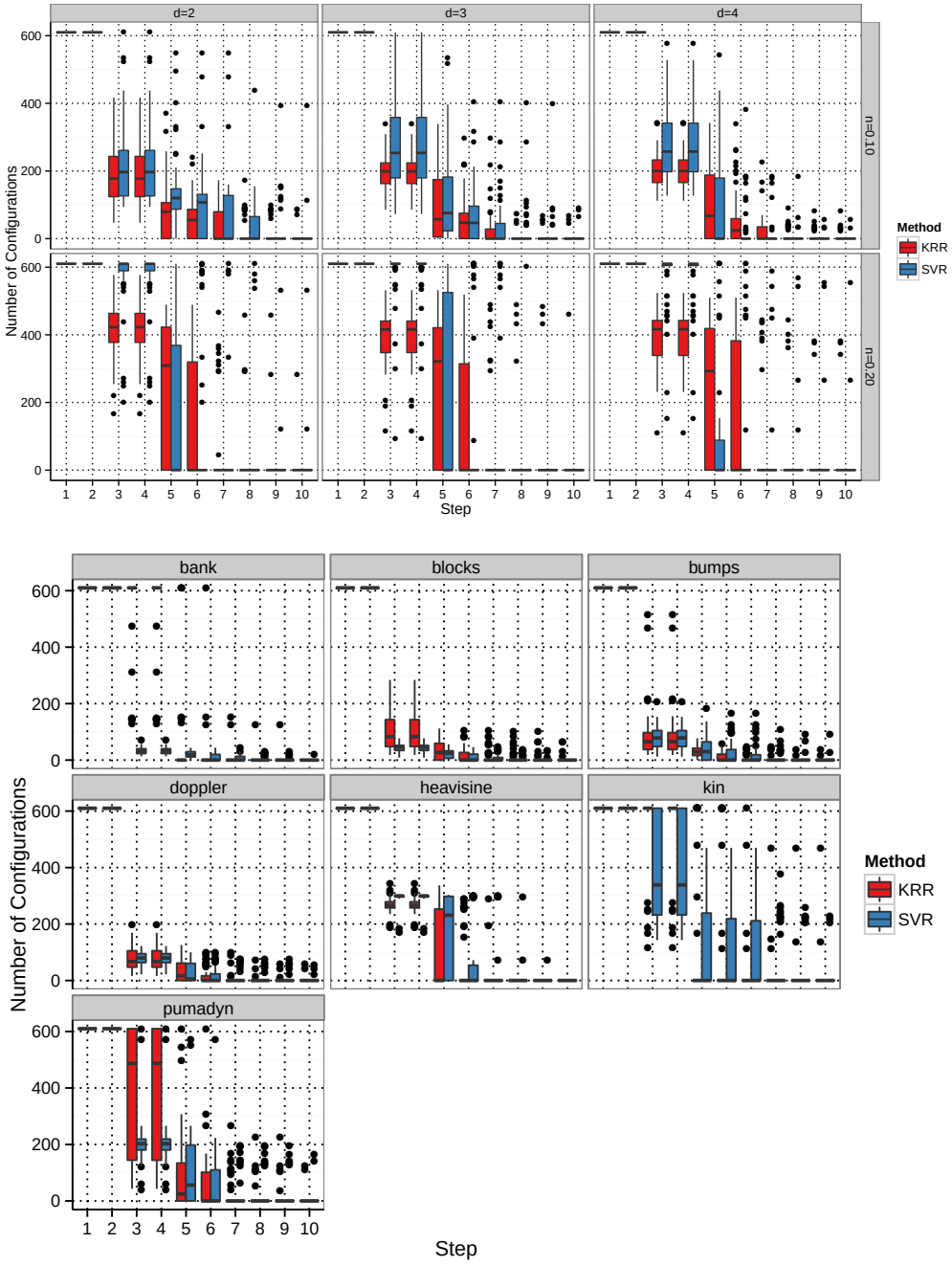


Figure 21: Remaining configurations after each step for the *noisy sinc* and different *benchmark* data sets using the outlier-based similarity test. Compared to Figure 13 and Figure 16 we can clearly observe the higher retention rate of this more conservative test.

with the same power it requires on average fewest observations for a decision. The testing scheme of Wald is called *open* since the procedure could potentially go on forever, as long as ℓ does not leave the (A, B)-tunnel.

The open design of Wald's procedure led to a development of a different kind of sequential tests, where the number of observations is fixed beforehand (see Armitage, 1960; Spicer, 1962; Alling, 1966; McPherson and Armitage, 1971). For instance in clinical studies it might be impossible or ethically prohibitive to use a test which potentially could go on forever. Unfortunately, none of these so-called closed tests exhibit an optimality criterion, therefore we choose one which at least in simulation studies showed the best behavior in terms of average sample number statistics: The method of Spicer (1962) is based on a gambler's ruin scenario in which both players have a fixed fortune and decide to play for n games. If $f(n, \pi, F_a, F_b)$ is the probability that a player with fortune F_a and stake b will ruin his opponent with fortune F_b in exactly n games, then the following recurrence holds:

$$f(n, \pi, F_a, F_b) = \begin{cases} 0 \quad \text{if } F_a < 0 \vee (n = 0 \wedge F_b > 0), \\ 1 \quad \text{if } n = 0 \wedge F_a > 0 \wedge F_b \leq 0, \\ \pi f(n-1, \pi, F_a + 1, F_b - b) \\ \qquad +(1-\pi) f(n-1, \pi, F_a - b, F_b + b) \quad \text{otherwise.} \end{cases}$$

In each step, the player can either win a game with probability π and win 1 from his opponent or lose the stake b to the other player. Now, given $n = x + y$ games of which player A has won y and player B has won x, the game will stop if either of the following conditions hold:

$$y - bx = -F_a \Leftrightarrow y = \frac{b}{1+b} n - \frac{F_a}{1+b} \quad \text{or} \quad y - bx = F_b \Leftrightarrow y = \frac{b}{1+b} n - \frac{F_b}{1+b}.$$

This formulation casts the gambler's ruin problem into a Wald-like scheme, where we just observe the cumulative wins of player A and check whether we reached the lower or upper line. If we now choose F_a and F_b such that $f(n, 0.5, F_a, F_b) \leq \alpha_l$, we construct a test which allows us to check whether a given configuration performs worse than $\pi = 0.5$ (i.e., crosses the lower line) and can therefore be flagged as an overall loser with controlled error probability of α_l (see Alling 1966). For more details on the closed design of Spicer please consult Spicer (1962).

Since simulation studies show that the closed variants of the sequential testing exhibit low average sample number statistics, we first have a look at the runtime performance of the CVST algorithm equipped with either the open or the closed sequential test. The most influential parameter in terms of runtime is the S parameter. In principle, a larger number of steps leads to more robust estimates, but also to an increase of computation time. We study the effect of different choices of this parameter in a simulation. For the sake of simplicity we assume that the binary top or flop scheme consists of independent Bernoulli variables with $\pi_{\text{winner}} \in [0.9, 1.0]$ and $\pi_{\text{loser}} \in [0.0, 0.1]$. We test both the open and the closed sequential test and compare the relative speed-up of the CVST algorithm compared to a full 10-fold cross-validation in case the learner is cubic.

Figure 22 shows the resulting simulated runtimes for different settings. The overall speed-up is much higher for the closed sequential test indicating a more aggressive behavior

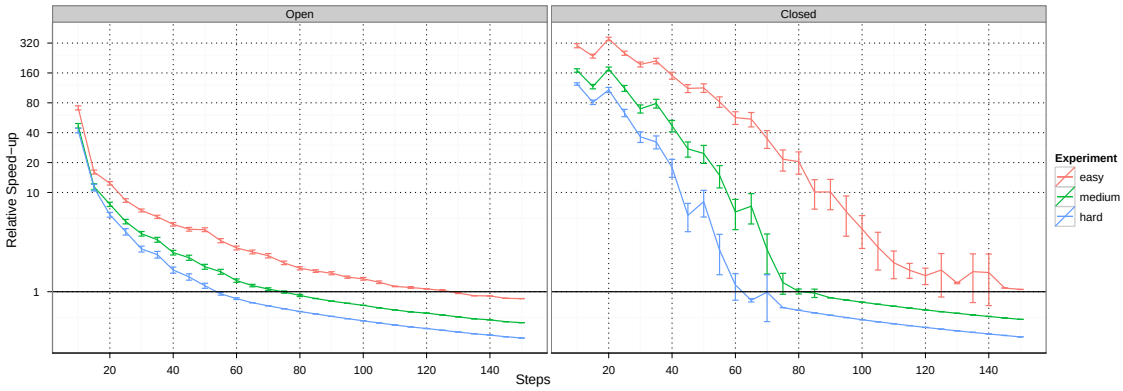


Figure 22: Relative speed gain of fast cross-validation compared to full cross-validation. We assume that training time is cubic in the number of samples. Shown are runtimes for 10-fold cross-validation on different problem classes by different loser/winner ratios (easy: 3:1; medium: 1:1, hard: 1:3) over 200 resamples.

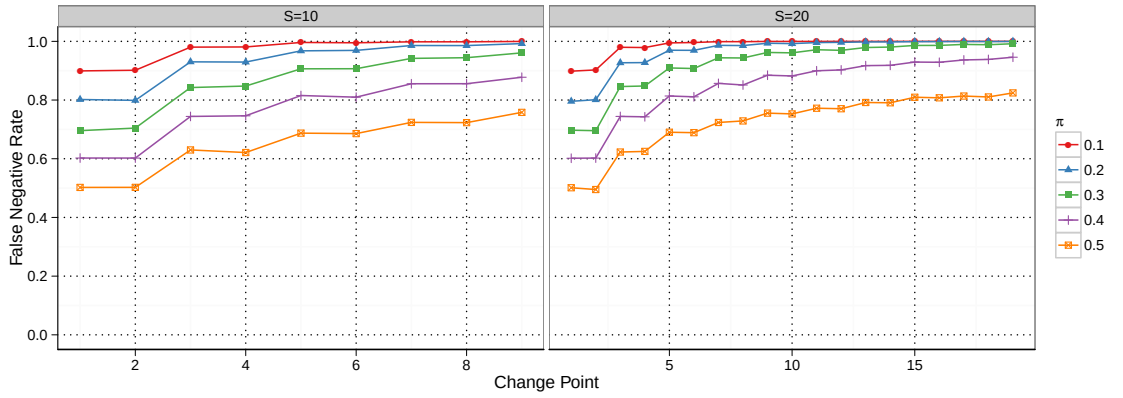


Figure 23: False negatives generated with the closed sequential test for non-stationary configurations, i.e., at the given change point the Bernoulli variable changes its π_{before} from the indicated value to 1.0.

compared to the more conservative open alternative. Both tests show their highest increase in the range of 10 to 20 steps with a rapid decline towards the higher step numbers. So in terms of speed the closed sequential test definitely beats the more conservative open test.

To evaluate the false negatives of the closed sequential test we simulate switching configurations by independent Bernoulli variables which change their success probability π from a chosen $\pi_{\text{before}} \in \{0.1, 0.2, \ldots, 0.5\}$ to a constant 1.0 at a given change point. By using this setup we mimic the behavior of a switching configuration which starts out as a loser

and after enough data is available turns into a constant winner. The results can be seen in Figure 23 which reveals that the speed gain comes at a price: Apart from having no control over the safety zone, the number of falsely dropped configurations is much higher than for the open sequential test (see Figure 7 in Section 5.1). While having a definitive advantage over the open test in terms of speed, the false negative rate of the closed test renders it useless for the CVST algorithm.

8. Conclusion

We presented a method to speed up the cross-validation procedure by starting at subsets of the full training set size, identifying clearly underperforming parameter configurations early on and focusing on the most promising candidates for the larger subset sizes. We have discussed that taking subsets of the data set has theoretical advantages when compared to other heuristics like local search on the parameter set because the effects on the test errors are systematic and can be understood statistically. On the one hand, we argued that the optimal configurations converge to the true ones as sample sizes tend to infinity, but we also discussed in a concrete setting how the different behaviors of estimation error and approximation error lead to much faster convergence practically. These insights led to the introduction of a safety zone through sequential testing, which ensures that underperforming configurations are not removed prematurely when the minima are not converged yet. In experiments we showed that our procedure leads to a speed-up of up to 120 times compared to the full cross-validation without a significant increase in prediction error.

It will be interesting to combine this method with other procedures like the Hoeffding races or algorithms for multi-armed bandit problems. Furthermore, getting accurate convergence bounds even for finite sample size settings is another topic for future research.

Acknowledgments

We would like to acknowledge support for this project from the BMBF project ALICE, "Autonomous Learning in Complex Environments" (01IB10003B). We would like to thank Klaus-Robert Müller, Marius Kloft, Raphael Pelossof and Ralf Herbrich for fruitful discussions and help. We would also like to thank the anonymous reviewers who have helped to clarify a number of points and further improve the paper.

Appendix A. Sample-Size Independent Parametrization

In Section 2 we needed as a precondition that the test performances converge for a fixed parameter configuration c as n tends to infinity. In this section, we discuss this condition in the context of the empirical risk minimization. We refer to the book by Devroye et al. (1996) and the references contained therein for the theoretical results.

In the empirical risk minimization framework, a learning algorithm is interpreted as choosing the solution g_n with the best error on the training set $\hat{R}_n(g) = \frac{1}{n}\sum_{i=1}^{n} \ell(g(X_i), Y_i)$ from some hypothesis class $\mathcal{G}$. If the VC-dimension of $\mathcal{G}$, which roughly measures the

complexity of $\mathcal{G}$, is finite then it holds that

$$\hat{R}_n(g) \to R(g)$$

uniformly over $g \in \mathcal{G}$, and consequently also $R(g_n) \to \inf_{g \in \mathcal{G}} R(g)$.

Now in order to make the link to our condition, we need that each parameter c corresponds to a fixed hypothesis class $\mathcal{G}_c$ (and not depend on the sample size in some way), and that the VC-dimension is finite. For feed-forward neural networks, one can show, for example, that neural networks with one hidden layer with k inner nodes and sigmoid activation function have finite VC-dimension (Devroye et al., 1996, Theorem 30.6).

For kernel machines, we consider the reproducing kernel Hilbert space (RKHS, Aronszajn 1950) view: Let $\mathcal{H}_k$ the RKHS induced by a Mercer kernel k with norm $\| \cdot \|_{\mathcal{H}_k}$. Evgeniou and Pontil (1999) show that the V_γ-dimension of the hypothesis class $\mathcal{G}(A) = \{f \in \mathcal{H}_k \mid \|f\|^2_{\mathcal{H}_k} \leq A\}$ is finite, from which uniform converges of the kind described above follows, and thus also that our condition holds.

Many kernel methods, including kernel ridge regression and support vector machines can be written as regularized optimization problems in the RKHS of the form:

$$\min_{f \in \mathcal{H}_k} \left(\frac{1}{n} \sum_{i=1}^{n} \ell(f(X_i), Y_i) + C\|f\|^2_{\mathcal{H}_k} \right) = \min_{f \in \mathcal{H}_k} \left(\hat{R}_n(f) + C\|f\|^2_{\mathcal{H}_k} \right).$$

Now if we assume that $\ell(f(x), y)$ is bounded by B and continuous in f, it follows that the minimum is attained for some f with $\|f\|^2_{\mathcal{H}_k} \leq B/C$: For $\|f\|_{\mathcal{H}_k} = 0$, $\hat{R}_n(f) + C\|f\|^2_{\mathcal{H}_k} \leq B$, and for $\|f\|_{\mathcal{H}_k} > B/C$, $\hat{R}_n(f) + C\|f\|^2_{\mathcal{H}_k} > B$. Because $\hat{R}(f) + C\|f\|^2_{\mathcal{H}_k}$ is continuous in f, it follows that the minimum is somewhere in-between.

Now $\hat{R}_n(f)$ converges to $R(f)$ uniformly over $f \in \mathcal{G}(B/C)$, such that there exists an $A \leq B/C$ such that

$$\min_{f \in \mathcal{H}_k} \left(\hat{R}(f) + C\|f\|^2_{\mathcal{H}_k} \right) = \min_{f \in \mathcal{G}(A)} R(f),$$

and we see that a regularization constant C corresponds to a fixed hypothesis class $\mathcal{G}(A)$ and our condition holds again.

As a direct consequence of this discussion we have to take care of the correct scaling of the regularization constants during the CVST run. Thus, for kernel ridge regression we have to scale the λ parameter linearly with the data set size and for the SVR divide the C parameter accordingly (see the reference implementation in the official CRAN package named `CVST` or the development version at `https://github.com/tammok/CVST` for details).

Appendix B. Example Run of CVST Algorithm

In this section we give an example of the whole CVST algorithm on one *noisy sinc* data set of $n = 1,000$ data points with intrinsic dimensionality of $d = 2$. The CVST algorithm is executed with $S = 10$ and $w_{\text{stop}} = 4$. We use a ν-SVM (Schölkopf et al., 2000) and test a parameter grid of $\log_{10}(\sigma) \in \{-3, -2.9, \ldots, 3\}$ and $\nu \in \{0.05, 0.1, \ldots, 0.5\}$. The procedure runs for 4 steps after which the early stopping rule takes effect. This yields the following traces matrix (only remaining configurations are shown):

	n = 90	**n = 180**	**n = 270**	**n = 360**
$\log_{10}(\sigma) = -2.3, \nu = 0.35$	0	0	1	0
$\log_{10}(\sigma) = -2.3, \nu = 0.40$	0	1	1	0
$\log_{10}(\sigma) = -2.3, \nu = 0.45$	0	1	0	1
$\log_{10}(\sigma) = -2.2, \nu = 0.30$	0	1	0	0
$\log_{10}(\sigma) = -2.2, \nu = 0.35$	0	1	1	0
$\log_{10}(\sigma) = -2.2, \nu = 0.40$	0	1	1	1
$\log_{10}(\sigma) = -2.2, \nu = 0.45$	0	1	1	1
$\log_{10}(\sigma) = -2.2, \nu = 0.50$	0	0	1	1
$\log_{10}(\sigma) = -2.1, \nu = 0.35$	0	1	1	1
$\log_{10}(\sigma) = -2.1, \nu = 0.40$	0	1	1	1
$\log_{10}(\sigma) = -2.1, \nu = 0.45$	0	1	1	1
$\log_{10}(\sigma) = -2.1, \nu = 0.50$	1	0	1	1
$\log_{10}(\sigma) = -2.0, \nu = 0.50$	0	0	1	1

The corresponding mean square errors of the remaining configurations after each step are shown in the next matrix. Based on these values, the winning configuration, namely $\log_{10}(\sigma) = -2.1, \nu = 0.40$ is chosen:

	n = 90	**n = 180**	**n = 270**	**n = 360**
$\log_{10}(\sigma) = -2.3, \nu = 0.35$	0.0370	0.0199	0.0145	0.0150
$\log_{10}(\sigma) = -2.3, \nu = 0.40$	0.0362	0.0197	0.0146	0.0146
$\log_{10}(\sigma) = -2.3, \nu = 0.45$	0.0356	0.0197	0.0146	0.0144
$\log_{10}(\sigma) = -2.2, \nu = 0.30$	0.0365	0.0195	0.0146	0.0148
$\log_{10}(\sigma) = -2.2, \nu = 0.35$	0.0351	0.0193	0.0142	0.0145
$\log_{10}(\sigma) = -2.2, \nu = 0.40$	0.0345	0.0194	0.0143	0.0141
$\log_{10}(\sigma) = -2.2, \nu = 0.45$	0.0340	0.0193	0.0143	0.0140
$\log_{10}(\sigma) = -2.2, \nu = 0.50$	0.0332	0.0200	0.0145	0.0138
$\log_{10}(\sigma) = -2.1, \nu = 0.35$	0.0353	0.0194	0.0144	0.0142
$\log_{10}(\sigma) = -2.1, \nu = 0.40$	0.0343	0.0195	0.0142	0.0138
$\log_{10}(\sigma) = -2.1, \nu = 0.45$	0.0340	0.0197	0.0140	0.0138
$\log_{10}(\sigma) = -2.1, \nu = 0.50$	0.0329	0.0199	0.0142	0.0137
$\log_{10}(\sigma) = -2.0, \nu = 0.50$	0.0351	0.0204	0.0145	0.0137

Appendix C. Nonparametric Tests

The tests used in the CVST algorithm are common tools in the field of statistical data analysis. Here we give a short summary based on Heckert and Filliben (2003) and cast the notation into the CVST framework context. Both methods deal with the performance matrix of K configurations with performance values on r data points:

	Data Points			
Configuration	1	2	...	r
1	x_{11}	x_{12}	...	x_{1r}
2	x_{21}	x_{22}	...	x_{2r}
3	x_{31}	x_{32}	...	x_{3r}
$\vdots$	$\vdots$	$\vdots$		$\vdots$
K	x_{K1}	x_{K2}	...	x_{Kr}

Both tests treat similar questions ("Do the K configurations have identical effects?") but are designed for different kinds of data: Cochran's Q test is tuned for binary x_{ij} while the Friedman test acts on continuous values. In the context of the CVST algorithm the tests are used for two different tasks:

1. Determine whether a set of configurations are the top performing ones (step ❶ in the overview Figure 3 and the function topConfigurations defined in Algorithm 2).

2. Check whether the remaining configurations behaved similar in the past (step ❸ in the overview Figure 3 and the function similarPerformance in Algorithm 4).

In both cases, the configurations are compared either by the performance on the samples (Point 1 above) or on the last w_{stop} traces (Point 2 above) of the remaining configurations. Depending on the learning problem either the Friedman test for regression task or the Cochran's Q test for classification tasks is used in Point 1.

In both cases the hypotheses for the tests are as follows:

- H_0: All configurations are equally effective (no effect)

- H_1: There is a difference in the effectiveness among the configurations, i.e., there is at least one configuration showing a significantly different effect on the data points.

C.1 Cochran's Q Test

The test statistic T is calculated as follows:

$$T = K(K-1)\frac{\sum_{i=1}^{K}(R_i - \frac{M}{K})^2}{\sum_{i=1}^{r} C_i(K - C_i)}$$

with R_i denoting the row total for the i^{th} configuration, C_i the column total for the i^{th} data point, and M the grand total. We reject H_0, if $T > \chi^2(1-\alpha, K-1)$ with $\chi^2(1-\alpha, K-1)$ denoting the $(1-\alpha)$-quantile of the χ^2 distribution with $K-1$ degrees of freedom and α is the significance level. As Cochran (1950) points out, the χ^2 approximation breaks down for small tables. Tate and Brown (1970) state that as long as the table contains at least 24 entries, the χ^2 approximation will suffice, otherwise the exact distribution should be used which can either be calculated explicitly (see Patil, 1975) or determined via permutation.

C.2 Friedman Test

Let $R(x_{ij})$ be the rank assigned to x_{ij} within data point i (i.e., rank of a configuration on data point i). Average ranks are used in the case of ties. The ranks for a configuration at position k are summed up over the data points to obtain

$$R_k = \sum_{i=1}^{r} R(x_{ki}).$$

The test statistic T is then calculated as follows:

$$T = \frac{12}{rK(K+1)} \sum_{i=1}^{K} (R_i - r(K+1)/2)^2.$$

If there are ties, then

$$T = \frac{(K-1)\sum_{i=1}^{K}(R_i - r(K+1)/2)^2}{[\sum_{i=1}^{K}\sum_{j=1}^{r} R(x_{ij})^2] - [rK(K+1)^2]/4}.$$

We reject H_0 if $T > \chi^2(\alpha, K-1)$ with $\chi^2(\alpha, K-1)$ denoting the α-quantile of the χ^2 distribution with $K-1$ degrees of freedom and α being the significance level.

Appendix D. Proof of Safety Zone Bound

In this section we prove the safety zone bound of Section 5.1 of the paper. We will follow the notation and treatment of the sequential analysis as found in the original publication of Wald (1947), Sections 5.3 to 5.5. First of all, Wald proves in Equation 5:27 that the following approximation holds:

$$\mathrm{ASN}(\pi_0, \pi_1 | \pi = 1.0) \approx \frac{\log \frac{1-\beta_l}{\alpha_l}}{\log \frac{\pi_1}{\pi_0}}.$$

where $\mathrm{ASN}(\cdot,\cdot)$ (Average Sample Number) is the expected number of steps until the given test will yield a decision, if the underlying success probability of the tested sequence is $\pi = 1.0$. The minimal $\mathrm{ASN}(\pi_0, \pi_1 | \pi = 1.0)$ is therefore attained if $\log \frac{\pi_1}{\pi_0}$ is maximal, which is clearly the case for $\pi_1 = 1.0$ and $\pi_0 = 0.5$, which holds by construction. So we get the lower bound of S for a given significance level α_l, β_l:

$$S \geq \left\lceil \log \frac{1-\beta_l}{\alpha_l} / \log 2 \right\rceil.$$

The lower line L_0 of the graphical sequential analysis test as exemplified in Figure 3 of the paper is defined as follows (see Equation 5:13 - 5:15):

$$L_0 = \frac{\log \frac{\beta_l}{1-\alpha_l}}{\log \frac{\pi_1}{\pi_0} - \log \frac{1-\pi_1}{1-\pi_0}} + n \frac{\log \frac{1-\pi_0}{1-\pi_1}}{\log \frac{\pi_1}{\pi_0} - \log \frac{1-\pi_1}{1-\pi_0}}.$$

Setting $L_0 = 0$, we can get the intersection of the lower test line with the x-axis and therefore the earliest step s_{safe}, in which the procedure will drop a constant loser configuration. This yields

$$s_{\text{safe}} = -\frac{\log\frac{\beta_l}{1-\alpha_l}}{\log\frac{\pi_1}{\pi_0} - \log\frac{1-\pi_1}{1-\pi_0}} \Big/ \frac{\log\frac{1-\pi_0}{1-\pi_1}}{\log\frac{\pi_1}{\pi_0} - \log\frac{1-\pi_1}{1-\pi_0}} = -\frac{\log\frac{\beta_l}{1-\alpha_l}}{\log\frac{1-\pi_0}{1-\pi_1}} = \frac{\log\frac{\beta_l}{1-\alpha_l}}{\log\frac{1-\pi_1}{1-\pi_0}} = \frac{\log\frac{\beta_l}{1-\alpha_l}}{\log 2 - \sqrt[S]{\frac{1-\beta_l}{\alpha_l}}}.$$

The last equality can be derived by inserting the closed form of π_1 given $\pi_0 = 0.5$:

$$S = \text{ASN}(\pi_0, \pi_1 | \pi = 1.0) = \frac{\log\frac{1-\beta_l}{\alpha_l}}{\log\frac{\pi_1}{\pi_0}} = \frac{\log\frac{1-\beta_l}{\alpha_l}}{\log 2\pi_1} \Leftrightarrow 2\pi_1 = \sqrt[S]{\frac{1-\beta_l}{\alpha_l}} \Leftrightarrow \pi_1 = \frac{1}{2}\sqrt[S]{\frac{1-\beta_l}{\alpha_l}}.$$

Setting s_{safe} in relation to the maximal number of steps S yields the safety zone bound of Section 5.1.

Appendix E. Proof of Computational Budget

For the size N of the whole data set and a learner of time complexity $f(n) = n^m$, where $m \in \mathbb{N}$, resulting in a learning time of $t = N^m$, one observes that learning on a proportion of size $\frac{i}{S}N$ takes about $\frac{i^m}{S^m}t$ time. By construction one has to learn on all K parameter configurations in each step before hitting $s_r \times S$ and on $K \times r$ parameter configurations with drop rate $(1-r)$ afterwards. Thus the entirely needed computation time is given by

$$K \times (1-r) \sum_{i=1}^{s_r \times S} \frac{i^m}{S^m} t + K \times r \sum_{i=1}^{S} \frac{i^m}{S^m} t$$

which should be smaller than the given time budget T.

Making use of the fact proved in Appendix E.1 that $\frac{1}{n^{m-1}} \sum_{i=1}^{n} i^m \dot{\leq} \frac{n^2}{m+1} + \frac{n}{2} + \frac{m}{12}$ holds under the mild condition of $n > \frac{m}{2\pi}$, where $\dot{\leq}$ describes an asymptotic relation, one can reformulate the inequality as follows:

$$\begin{aligned} &\frac{tK(1-r)s_r^{m-1}}{S} \frac{1}{(s_r S)^{m-1}} \sum_{i=1}^{s_r S} i^m + \frac{tKr}{S} \frac{1}{S^{m-1}} \sum_{i=1}^{S} i^m \\ \dot{\leq}\ &\frac{tK}{S}\left[(1-r)s_r^{m-1}\left(\frac{(s_r S)^2}{m+1} + \frac{s_r S}{2} + \frac{m}{12}\right) + r\left(\frac{S^2}{m+1} + \frac{S}{2} + \frac{m}{12}\right)\right] \dot{\leq} T. \end{aligned}$$

It is obvious that this inequality is quadratic in the variable S which can be solved by bringing the above inequality in standard form:

$$\begin{aligned} 0 \dot{\geq}\ &\left[(1-r)s_r^{m-1}\left(\frac{(s_r S)^2}{m+1} + \frac{s_r S}{2} + \frac{m}{12}\right) + r\left(\frac{S^2}{m+1} + \frac{S}{2} + \frac{m}{12}\right)\right] - \frac{TS}{tK} \\ \Leftrightarrow\ 0 \dot{\geq}\ &\frac{(1-r)s_r^{m+1} + r}{m+1} S^2 + \left[\frac{(1-r)s_r^m + r}{2} - \frac{T}{tK}\right] S + \left((1-r)s_r^{m-1} + r\right)\frac{m}{12} \\ \Leftrightarrow\ 0 \dot{\geq}\ &S^2 + 2\left[\frac{m+1}{4}\frac{tK(1-r)s_r^m + tKr - 2T}{((1-r)s_r^{m+1} + r)tK}\right] S + \frac{m(m+1)}{12}\frac{(1-r)s_r^{m-1} + r}{(1-r)s_r^{m+1} + r}. \end{aligned}$$

Substituting $a = \frac{m+1}{4}\frac{tK(1-r)s_r^m+tKr-2T}{((1-r)s_r^{m+1}+r)tK}$ and $b = \frac{m(m+1)}{12}\frac{(1-r)s_r^{m-1}+r}{(1-r)s_r^{m+1}+r}$ above is equivalent to:

$$S = -a + y,\ y \in \left\{-\sqrt{a^2-b}, +\sqrt{a^2-b}\right\}.$$

For the sake of a meaningful step amount, i.e., $S > 0$ and furthermore S as large as possible we choose it as

$$S = \left\lfloor -a + \sqrt{a^2-b} \right\rfloor.$$

Note that S is a function of the parameter s_r. The mild condition for the upper bound of power sums mentioned above has to be fulfilled. Since obviously $b \geq 0$ holds, a must be negative in order to gain a positive step amount. Furthermore the root has to be solvable. So the following constraints on s_r have to be made:

$$\begin{aligned} &(1) \qquad 2T \geq tK(1-r)s_r^m + tKr \\ &(2) \qquad a^2 \geq b \\ &(3) \qquad s_r S > \frac{m}{2\pi}. \end{aligned}$$

Note that condition (3) is trivial for a small degree of complexity m, which is the common case.

E.1 Proof of the Upper Bound

Assume that $n = \frac{m}{c}$ where $c < 2\pi$. Denote by B_i the *Bernoulli* numbers.

$$\begin{aligned} \frac{1}{n^{m-1}}\sum_{i=1}^{n} i^m &= \frac{1}{n^{m-1}}\Big[\frac{n^m}{2} + \frac{1}{m+1}\sum_{k=0}^{\lfloor\frac{m}{2}\rfloor}\binom{m+1}{2k}B_{2k}n^{m+1-2k}\Big] \\ &= \frac{n^2}{m+1} + \frac{n}{2} + \frac{m}{12} + \sum_{k=2}^{\lfloor\frac{m}{2}\rfloor}(-1)^{i+1}\frac{1}{m+1}\binom{m+1}{2k}|B_{2k}|n^{2-2k} \end{aligned}$$

The sum term is alternating in sign and asymptotically monotone decreasing in k:

$$\begin{aligned} \frac{1}{m+1}\binom{m+1}{2k}|B_{2k}|n^{2-2k} &\sim \frac{m!}{(2k)!(m+1-2k)!}2\frac{(2k)!}{(2\pi)^{2k}}\left(\frac{m}{c}\right)^{2-2k} \\ &= \frac{2m}{c^2}\underbrace{\prod_{j=0}^{2k-2}\left(1-\frac{j}{m}\right)}_{\downarrow 0 \text{ as } k\to\infty}\underbrace{\left(\frac{c}{2\pi}\right)^{2k}}_{\downarrow 0,\ \frac{c}{2\pi}<1} \end{aligned}$$

where we use the asymptotic behavior of $|B_{2k}| \sim 2\frac{(2k)!}{(2\pi)^{2k}}$. Now that the sequence under the sum starts negative, grouping up each two subsequent elements gives a negative value, such that the sum also is negative.

Therefore

$$\frac{1}{n^{m-1}}\sum_{i=1}^{n} i^m \dot{\leq} \frac{n^2}{m+1} + \frac{n}{2} + \frac{m}{12}.$$

References

D. W. Alling. Closed sequential tests for binomial probabilities. *Biometrika*, 53(1/2):73–84, 1966.

S. Arlot, A. Celisse, and P. Painleve. A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4:40–79, 2010.

P. Armitage. *Sequential Medical Trials.* Blackwell Scientific Publications, 1960.

N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68:337–404, 1950.

P. L. Bartlett, P. M. Long, and R. C. Williamson. Fat-shattering and the learnability of real-valued functions. *Journal of Computer and Systems Science*, 52:434–452, 1996.

Y. Bengio. Gradient-based optimization of hyperparameters. *Neural Computation*, 12(8): 1889–1900, 2000.

J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13:281–305, March 2012.

J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems 24*, pages 2546–2554, 2011.

D. Berry and B. Fristedt. *Bandit problems: Sequential Allocation of Experiments.* Chapman & Hall, 1985.

M. Birattari. *Tuning Metaheuristics: A Machine Learning Perspective.* Springer, 2009.

M. Birattari, T. Stützle, L. Paquete, and K. Varrentrapp. A racing algorithm for configuring metaheuristics. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 11–18, 2002.

J. A. Blackard and D. J. Dean. Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. *Computers and Electronics in Agriculture*, 24:131–151, 1999.

J. K. Bradley and R. Schapire. Filterboost: Regression and classification on large datasets. In *Advances in Neural Information Processing Systems 20*, pages 185–192, 2008.

M. L. Braun, J. Buhmann, and K.-R. Müller. On relevant dimensions in kernel feature spaces. *Journal of Machine Learning Research*, 9:1875–1908, 2008.

N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games.* Cambridge University Press, 2006.

S. Chien, J. Gratch, and M. Burl. On the efficient allocation of resources for hypothesis evaluation: A statistical approach. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 17:652–665, 1995.

S. Chien, A. Stechert, and D. Mutz. Efficient heuristic hypothesis ranking. *Journal of Artificial Intelligence Research*, 10:375–397, 1999.

W. G. Cochran. The comparison of percentages in matched samples. *Biometrika*, 37(3-4): 256–266, 1950.

L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer, 1996.

P. Domingos and G. Hulten. A general method for scaling up machine learning algorithms and its application to clustering. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 106–113, 2001.

D. L. Donoho and J. M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994.

T. Evgeniou and M. Pontil. On the V gamma dimension for regression in reproducing kernel hilbert spaces. In O. Watanabe and T. Yokomori, editors, *Algorithmic Learning Theory*, pages 106–117, 1999.

M. Friedman. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 32(200):675–701, 1937.

S. Geisser. The predictive sample reuse method with applications. *Journal of the American Statistical Association*, 70(350):320–328, 1975.

N. A. Heckert and J. J. Filliben. *NIST Handbook 148: DATAPLOT Reference Manual, Volume I: Commands*. National Institute of Standards and Technology Handbook Series, 2003.

V. Heidrich-Meisner and C. Igel. Hoeffding and Bernstein races for selecting policies in evolutionary direct policy search. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 401–408, 2009.

S. S. Keerthi, V. Sindhwani, and O. Chapelle. An efficient method for gradient-based adaptation of hyperparameters in SVM models. In *Advances in Neural Information Processing Systems 19*, pages 673–680, 2006.

R. Kohavi and G. H. John. Automatic parameter selection by minimizing estimated error. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 304–312, 1995.

W. H. Kruskal and W. A. Wallis. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 47(260):583–621, 1952.

O. Maron and A. W. Moore. Hoeffding races: Accelerating model selection search for classification and function approximation. In *Advances in Neural Information Processing Systems 6*, pages 59–66, 1994.

O. Maron and A. W. Moore. The racing algorithm: Model selection for lazy learners. *Artificial Intelligence Review*, 11:193–225, 1997.

C. K. McPherson and P. Armitage. Repeated significance tests on accumulating data when the null hypothesis is not true. *Journal of the Royal Statistical Society. Series A*, 134(1): 15–25, 1971.

V. Mnih, C. Szepesvári, and J.-Y. Audibert. Empirical Bernstein stopping. In *Proceedings of the 25th International Conference on Machine learning*, ICML '08, pages 672–679, 2008.

M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations Of Machine Learning*. MIT Press, 2012.

F. Mosteller and J. W. Tukey. *Handbook of Social Psychology*, volume 2, chapter Data analysis, including statistics, pages 80–203. Addison-Wesley, 2nd edition, 1968.

K. D. Patil. Cochran's Q test: Exact distribution. *Journal of the American Statistical Association*, 70(349):186–189, 1975.

R. Pelossof and M. Jones. Curtailed online boosting. Technical report, Columbia University, 2009.

R. Pelossof and Z. Ying. The attentive perceptron. *Computing Research Repository*, abs/1009.5972, 2010.

R. Pelossof and Z. Ying. Rapid learning with stochastic focus of attention. *Computing Research Repository*, abs/1105.0382, 2011.

G. Rätsch, T. Onoda, and K.-R. Müller. Soft margins for AdaBoost. *Machine Learning*, 42 (3):287–320, 2001.

V. Roth. Probabilistic discriminative kernel classifiers for multi-class problems. In *Proceedings of the 23rd DAGM-Symposium on Pattern Recognition*, pages 246–253, 2001.

B. Schölkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett. New support vector algorithms. *Neural Computation*, 12:1207–1245, 2000.

S. Smale and D.-X. Zhou. Estimating the approximation error in learning theory. *Analysis and Applications*, 1(1):1–25, 2003.

J. Snoek, H. Larochelle, and R. P. Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems 25*, pages 2960–2968, 2012.

C. C. Spicer. Some new closed sequential designs for clinical trials. *Biometrics*, 18(2): 203–211, 1962.

A. Stanski. *Konstruktives Probabilistisches Lernen*. PhD thesis, Technische Universität Berlin, 2012.

I. Steinwart and C. Scovel. Fast rates for support vector machines using Gaussian kernels. *Annals of Statistics*, 35:575–607, 2007.

M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B*, 36(2):111–147, 1974.

M. W. Tate and S. M. Brown. Note on the Cochran Q test. *Journal of the American Statistical Association*, 65(329):155–160, 1970.

C. Thornton, F. Hutter, H. H. Hoos, and K. Leyton-Brown. Auto-WEKA: Automated selection and hyper-parameter optimization of classification algorithms. *CoRR*, abs/1208.3719, 2012.

V. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

A. Wald. *Sequential Analysis*. Wiley, 1947.

A. Wald and J. Wolfowitz. Optimum character of the sequential probability ratio test. *The Annals of Mathematical Statistics*, 19(3):326–339, 1948.

G. B. Wetherill and K. D. Glazebrook. *Sequential Methods in Statistics*. Chapman and Hall, 1986.

E. B. Wilson. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212, 1927.

A. X. Zheng and M. Bilenko. Lazy paired hyper-parameter tuning. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, pages 1924–1931, 2013.

Journal of Machine Learning Research 16 (2015) 1157-1210 Submitted 12/12; Revised 8/14; Published 6/15

Learning the Structure and Parameters of Large-Population Graphical Games from Behavioral Data

Jean Honorio JHONORIO@CSAIL.MIT.EDU
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139, USA

Luis Ortiz LEORTIZ@CS.STONYBROOK.EDU
Department of Computer Science
Stony Brook University
Stony Brook, NY 11794-4400, USA

Editor: Shie Mannor

Abstract

We consider learning, from *strictly* behavioral data, the structure and parameters of *linear influence games (LIGs)*, a class of parametric graphical games introduced by Irfan and Ortiz (2014). LIGs facilitate *causal strategic inference (CSI)*: Making inferences from causal interventions on stable behavior in strategic settings. Applications include the identification of the most influential individuals in large (social) networks. Such tasks can also support policy-making analysis. Motivated by the computational work on LIGs, we cast the learning problem as maximum-likelihood estimation (MLE) of a generative model defined by *pure-strategy Nash equilibria (PSNE)*. Our simple formulation uncovers the fundamental interplay between goodness-of-fit and model complexity: good models capture equilibrium behavior within the data while controlling the true number of equilibria, including those unobserved. We provide a generalization bound establishing the sample complexity for MLE in our framework. We propose several algorithms including *convex loss minimization (CLM)* and sigmoidal approximations. We prove that the number of exact PSNE in LIGs is small, with high probability; thus, CLM is sound. We illustrate our approach on synthetic data and real-world U.S. congressional voting records. We briefly discuss our learning framework's generality and potential applicability to general graphical games.

Keywords: linear influence games, graphical games, structure and parameter learning, behavioral data in strategic settings

1. Introduction

Game theory has become a central tool for modeling multi-agent systems in AI. Non-cooperative game theory has been considered as the appropriate mathematical framework in which to formally study *strategic* behavior in multi-agent scenarios.[1] The core solution concept of *Nash equilibrium (NE)* (Nash, 1951) serves a descriptive role of the stable outcome of the overall behavior of systems involving self-interested individuals interacting strategically with each other in distributed settings for which no direct global control is

1. See, e.g., the survey of Shoham (2008) and the books of Nisan et al. (2007) and Shoham and Leyton-Brown (2009) for more information.

possible. NE is also often used in predictive roles as the basis for what one might call *causal strategic inference*, i.e., inferring the results of causal interventions on stable actions/behavior/outcomes in strategic settings (See, e.g., Ballester et al. 2004, 2006; Heal and Kunreuther 2003, 2006, 2007; Kunreuther and Michel-Kerjan 2007; Ortiz and Kearns 2003; Kearns 2005; Irfan and Ortiz 2014, and the references therein). Needless to say, the computation and analysis of NE in games is of significant interest to the computational game-theory community within AI.

The introduction of compact representations to game theory over the last decade have extended computational/algorithmic game theory's potential for large-scale, practical applications often encountered in the real-world. For the most part, such game model representations are analogous to probabilistic graphical models widely used in machine learning and AI.[2] Introduced within the AI community about a decade ago, *graphical games* (Kearns et al., 2001) constitute an example of one of the first and arguably one of the most influential graphical models for game theory.[3]

There has been considerable progress on problems of *computing* classical equilibrium solution concepts such as NE and *correlated equilibria (CE)* (Aumann, 1974) in graphical games (see, e.g., Kearns et al. 2001; Vickrey and Koller 2002; Ortiz and Kearns 2003; Blum et al. 2006; Kakade et al. 2003; Papadimitriou and Roughgarden 2008; Jiang and Leyton-Brown 2011 and the references therein). Indeed, graphical games played a prominent role in establishing the computational complexity of computing NE in general normal-form games (see, e.g., Daskalakis et al. 2009 and the references therein).

An example of a recent computational application of non-cooperative game-theoretic graphical modeling and *causal strategic inference (CSI)* that motivates the current paper is the work of Irfan and Ortiz (2014). They proposed a new approach to the study of influence and the identification of the "most influential" individuals (or nodes) in large (social) networks. Their approach is strictly game-theoretic in the sense that it relies on non-cooperative game theory and the central concept of *pure-strategy Nash equilibria (PSNE)*[4] as an approximate *predictor* of *stable behavior* in *strategic* settings, and, unlike other models of behavior in mathematical sociology,[5] it is not interested and thus avoids explicit modeling of the complex dynamics by which such stable outcomes could have arisen or could be achieved. Instead, it concerns itself with the "bottom-line" end-state stable outcomes (or steady state behavior). Hence, the proposed approach provides an alternative to models based on the diffusion of behavior through a social network (See Kleinberg 2007 for an introduction and discussion targeted to computer scientists, and further references).

2. The fundamental property such compact representation of games exploit is that of *conditional independence*: each player's payoff function values are determined by the actions of the player and those of the player's neighbors only, and thus are *conditionally (payoff) independent* of the actions of the non-neighboring players, *given* the action of the neighboring players.
3. Other game-theoretic graphical models include *game networks* (La Mura, 2000), *multi-agent influence diagrams (MAIDs)* (Koller and Milch, 2003), and *action-graph games* (Jiang and Leyton-Brown, 2008).
4. In this paper, because we concern ourselves primarily with PSNE, whenever we use the term "equilibrium" or "equilibria" without qualification, we mean PSNE.
5. Some of these models have recently gained interest and have been studied within computer science, specially those related to *diffusion or contagion processes* (see, e.g., Granovetter 1978; Morris 2000; Domingos and Richardson 2001; Domingos 2005; Even-Dar and Shapira 2007).

The underlying assumption for most work in computational game theory that deals with algorithms for computing equilibrium concepts is that the games under consideration are already available, or have been "hand-designed" by the analyst. While this may be possible for systems involving a handful of players, it is in general impossible in systems with at least tens of agent entities, if not more, as we are interested in this paper.[6] For instance, in their paper, Irfan and Ortiz (2014) propose a class of games, called *influence games*. In particular, they concentrate on *linear influence games (LIGs)*, and, as briefly mentioned above, study a variety of *computational* problems resulting from their approach, *assuming such games are given as input.*

Research in computational game theory has paid relatively little attention to the problem of *learning* (both the structure and parameters of) graphical games from data. Addressing this problem is essential to the development, potential use and success of game-theoretic models in practical applications. Indeed, we are beginning to see an increase in the availability of data collected from processes that are the result of deliberate actions of agents in complex system. A lot of this data results from the interaction of a large number of individuals, being not only people (i.e., individual human decision-makers), but also companies, governments, groups or engineered autonomous systems (e.g., autonomous trading agents), for which any form of global control is usually weak. The Internet is currently a major source of such data, and the smart grid, with its trumpeted ability to allow individual customers to install autonomous control devices and systems for electricity demand, will likely be another one in the near future.

In *this* paper, we investigate in considerable technical depth the problem of *learning LIGs from strictly behavioral data*: We do not assume the availability of utility, payoff or cost information in the data; the problem is *precisely* to infer that information from just the joint behavior collected in the data, *up to the degree needed to explain the joint behavior itself. We expect that, in most cases, the parameters quantifying a utility function or best-response condition are unavailable and hard to determine in real-world settings.* The availability of data resulting from the observation of an individual *public behavior* is arguably a weaker assumption than the availability of individual *utility* observations, which are often *private*. In addition, *we do not assume prior knowledge of the conditional payoff/utility independence structure as represented by the game graph.*

Motivated by the work of Irfan and Ortiz (2014) on a strictly non-cooperative game-theoretic approach to influence and strategic behavior in networks, we present a formal framework and design algorithms for learning the structure and parameters of LIGs with a *large* number of players. We concentrate on data about what one might call "the bottom line:" i.e., data about"end-states", "steady-states" or final behavior as represented by *possibly noisy* samples of joint actions/pure-strategies from stable outcomes, which we assume come from a *hidden* underlying game. Thus, we do not use, consider or assume available any *temporal* data about the detailed behavioral *dynamics*. In fact, the data we consider does not contain the dynamics that might have possibly led to the potentially stable joint-action

6. Of course, modeling and hand-crafting games for systems with many agents may be possible if the system has particular structure one could exploit. To give an example, this would be analogous to how one can exploit the probabilistic structure of HMMs to deal with long stochastic processes in a representationally succinct and computationally tractable way. Yet, we believe it is fair to say that such systems are largely/likely the exception in real-world settings in practice.

outcome! Since scalability is one of our main goals, we aim to propose methods that are polynomial-time in the number of players.

Given that LIGs belong to the class of 2-action graphical games (Kearns et al., 2001) with *parametric* payoff functions, we first needed to deal with the relative dearth of work on the broader problem of learning *general graphical games* from purely behavioral data. Hence, in addressing this problem, while inspired by the computational approach of Irfan and Ortiz (2014), the learning problem formulation we propose is in principle applicable to arbitrary games (although, again, the emphasis is on the PSNE of such games). In particular, we introduce a simple statistical generative mixture model, built "on top of" the game-theoretic model, with the only objective being to capture noise in the data. Despite the simplicity of the generative model, we are able to learn games from U.S. congressional voting records, which we use as a source of real-world behavioral data, that, as we will illustrate, seem to capture interesting, non-trivial aspects of the U.S. congress. While such models learned from real-world data are impossible to validate, we argue that there exists a considerable amount of anecdotal evidence for such aspects as captured by the models we learned. Figure 1 provides a brief illustration. (Should there be further need for clarification as to the why we present this figure, please see Footnote 14.)

As a final remark, given that LIGs constitute a non-trivial sub-class of parametric graphical games, we view our work as a step in the direction of addressing the broader problem of learning general graphical games with a large number of players from strictly behavioral data. We also hope our work helps to continue to bring and increase attention from the machine-learning community to the problem of inferring games from behavioral data (in which we attempt to *learn a game* that would "rationalize" players' observed behavior).[7]

1.1 A Framework for Learning Games: Desiderata

The following list summarizes the discussion above and guides our choices in our pursuit of a machine-learning framework for learning game-theoretic graphical models from strictly behavioral data.

- The learning algorithm
  - must output an LIG (which is a special type of graphical game); and
  - should be practical and tractably deal with a *large* number of players (typically in the hundreds, and certainly at least 4).
- The learned model objective is the "bottom line" in the sense that the basis for its evaluation is the prediction of *end-state* (or steady-state) joint decision-making behavior, and *not* the temporal behavioral *dynamics* that might have lead to end-state or the stable steady-state joint behavior.[8]

7. This is a type of problem arising from game theory and economics that is different from the problem of learning *in* games (in which the focus is the study of how individual players *learn to play* a game by a sequence of repeated interactions), a more matured and perhaps better known problem within machine learning (see, e.g., Fudenberg and Levine 1999).
8. Note that we are in no way precluding dynamic models as a way to end-state prediction. But there is no inherent need to make any *explicit* attempt or effort to model or predict the temporal behavioral *dynamics* that might have lead to end-state or the stable steady-state joint behavior, including pre-play "cheap talk," which are often overly complex processes. (See Appendix A.1 for further discussion.)

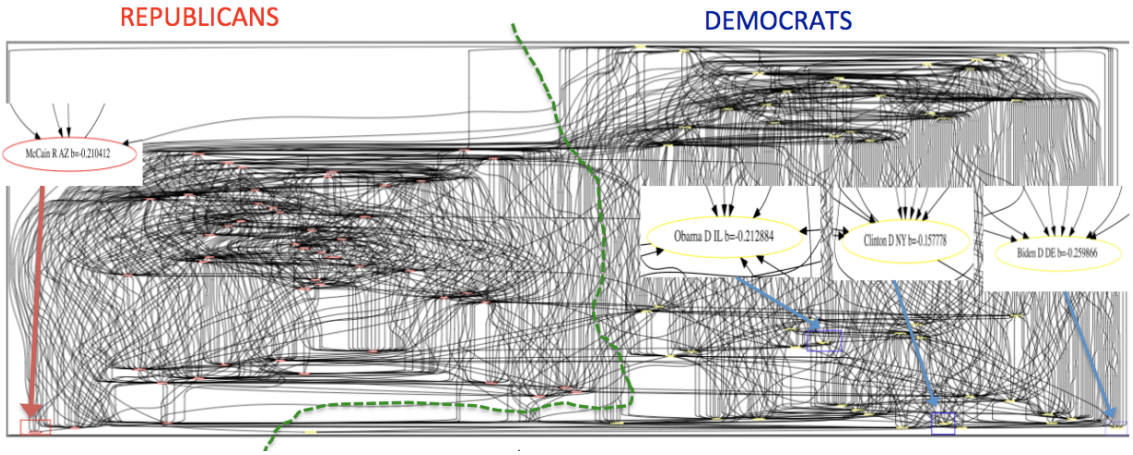


Figure 1: **110th US Congress's LIG (January 3, 2007-09):** We provide an illustration of the application of our approach to real congressional voting data. Irfan and Ortiz (2014) use such games to address a variety of computational problems, including the identification of *most influential* senators. (We refer the reader to their paper for further details.) We show the graph connectivity of a game learned by independent ℓ_1-regularized logistic regression (see Section 6.5). The reader should focus on the overall characteristics of the graph and not the details of the connectivity or the actual "influence" weights between senators. We highlight some particularly interesting characteristics consistent with anecdotal evidence. First, senators are more likely to be influenced by members of the same party than by members of the opposite party (the dashed green line denotes the separation between the parties). Republicans were "more strongly united" (tighter connectivity) than Democrats at the time. Second, the current US Vice President Biden (Dem./Delaware) and McCain (Rep./Arizona) are displayed at the "extreme of each party" (Biden at the bottom-right corner, McCain at the bottom-left) eliciting their opposite ideologies. Third, note that Biden, McCain, the current US President Obama (Dem./Illinois) and US Secretary of State Hillary Clinton (Dem./New York) have very few outgoing arcs; e.g., Obama only directly influences Feingold (Dem./Wisconsin), a prominent senior member with strongly liberal stands. One may wonder why do such prominent senators seem to have so little direct influence on others? A possible explanation is that US President Bush was about to complete his second term (the maximum allowed). Both parties had *very long* presidential primaries. All those senators contended for the presidential candidacy within their parties. Hence, one may posit that those senators were focusing on running their campaigns and that their influence in the *day-to-day* business of congress was channeled through other prominent senior members of their parties.

- The learning framework
  - would *only* have available strictly behavioral data on actual decisions/actions taken. It cannot require or use any kind of payoff-related information.
  - should be agnostic as to the type or nature of the decision-maker and does not assume each player is a *single human*. Players can be institutions or governments, or associated with the decision-making process of a group of individuals representing, e.g., a company (or sub-units, office sites within a company, etc.), a nation state (like in the UN, NATO, etc.), or a voting district. In other words, the recorded behavioral actions of each player may really be a representative of larger entities or groups of individuals, not necessarily a single human.
  - must provide computationally efficient learning algorithm with provable guarantees: worst-case polynomial running time in the number of players.
  - should be "data efficient" and provide provable guarantees on sample complexity (given in terms of "generalization" bounds).

1.2 Technical Contributions

While our probabilistic model is inspired by the concept of equilibrium from game theory, our technical contributions are not in the field of game theory nor computational game theory. Our technical contributions and the tools that we use are the ones in classical machine learning.

Our technical contributions include a novel generative model of behavioral data in Section 4 for general games. Motivated by the LIGs and the computational game-theoretic framework put forward by Irfan and Ortiz (2014), we formally define "identifiability" and "triviality" within the context of non-cooperative graphical games based on PSNE as the solution concept for stable outcomes in large strategic systems. We provide conditions that ensure identifiability among non-trivial games. We then present the maximum-likelihood estimation (MLE) problem for general (non-trivial identifiable) games. In Section 5, we show a generalization bound for the MLE problem as well as an upper bound of the functional/strategic complexity (i.e., analogous to the"VC-dimension" in supervised learning) of LIGs. In Section 6, we provide technical evidence justifying the *approximation* of the original problem by maximizing the number of *observed* equilibria in the data as suitable for a hypothesis-space of games with small *true* number of equilibria. We then present our convex loss minimization approach and a baseline sigmoidal approximation for LIGs. For completeness, we also present exhaustive search methods for both general games as well as LIGs. In Section 7, we formally define the concept of *absolute-indifference* of players and show that our convex loss minimization approach produces games in which all players are *non-absolutely-indifferent*. We provide a bound which shows that LIGs have small *true* number of equilibria with high probability.

2. Related Work

We provide a brief summary overview of previous work on learning games here, and delay discussion of the work presented below until after we formally present our model; this will provide better context and make "comparing and contrasting" easier for those interested,

Reference	Class	Needs Payoff	Learns Param.	Learns Struct.	Guarant.	Equil. Concept	Dyn.	Num. Agents
Wright and Leyton-Brown (2010)	NF	Y	N[a]	-	N	QRE	N	2
Wright and Leyton-Brown (2012)	NF	Y	N[a]	-	N	QRE	N	2
Gao and Pfeffer (2010)	NF	Y	Y	-	N	QRE	N	2
Vorobeychik et al. (2007)	NF	Y	Y	-	N	MSNE	N	2-5
Ficici et al. (2008)	NF	Y	Y	-	N	MSNE	N	10-200
Duong et al. (2008)	NGT	Y	N[a]	N	N	-	N	4,10
Duong et al. (2010)	NGT[b]	Y	N[c]	N	N	-	Y[d]	10
Duong et al. (2012)	NGT[b]	Y	N[c]	Y[e]	N	-	Y[d]	36
Duong et al. (2009)	GG	Y	Y	Y[f]	N	PSNE	N	2-13
Kearns and Wortman (2008)	NGT	N	-	-	Y	-	Y	100
Ziebart et al. (2010)	NF	N	Y	-	N	CE	N	2-3
Waugh et al. (2011)	NF	N	Y	-	Y	CE	Y	7
Our approach	***GG***	***N***	***Y***	***Y***[g]	***Y***	***PSNE***	***N***	***100***[g]

Table 1: **Summary of approaches for learning models of behavior.** See main text for a discussion. For each method we show its model class (GG: graphical games, NF: normal-form non-graphical games, NGT: non-game-theoretic model); whether it needs observed payoffs, learns *utility* parameters, learns *graphical* structure or provides guarantees(e.g., generalization, sample complexity or PAC learnability); its equilibria concept (PSNE: pure strategy or MSNE: mixed strategy Nash equilibria, CE: correlated equilibria, QRE: quantal response equilibria), whether it is dynamic (i.e., behavior predicted from past behavior); and the number of agents in the experimental validation. Note that there are relatively fewer models that do not assume observed payoff; among them, our method is the only one that learns the structure of graphical games, furthermore, we provide guarantees and a polynomial-time algorithm. [a]Learns only the "rationality parameter". [b]A graphical game could in principle be extracted, after removing the temporal/dynamic part. [c]It learns parameters for the "potential functions." [d]If the dynamic part is kept, it is not a graphical game. [e]It performs greedy search by constraining the maximum degree. [f]It performs branch and bound. [g]It has polynomial time-complexity in the number of agents, thus it can scale to thousands.

without affecting expert readers who may want to get to the technical aspects of the paper without much delay.

Table 1 constitutes our best attempt at a simple visualization to fairly present the differences and similarities of previous approaches to modeling behavioral data within the computational game-theory community in AI.

The research interest of previous work varies in what they intend to capture in terms of different aspects of behavior (e.g., dynamics, probabilistic vs. strategic) or simply different settings/domains (i.e., modeling "real human behavior," knowledge of achieved payoff or utility, etc.).

With the exception of Ziebart et al. (2010); Waugh et al. (2011); Kearns and Wortman (2008), previous methods assume that the actions as well as corresponding payoffs (or noisy samples from the true payoff function) are *observed* in the data. Our setting largely differs

from Ziebart et al. (2010); Kearns and Wortman (2008) because of their focus on system dynamics, in which future behavior is predicted from a sequence of past behavior. Kearns and Wortman (2008) proposed a learning-theory framework to model *collective* behavior based on *stochastic* models.

Our problem is clearly different from methods in *quantal response models* (McKelvey and Palfrey, 1995; Wright and Leyton-Brown, 2010, 2012) and *graphical multiagent models (GMMs)* (Duong et al., 2008, 2010) that assume *known structure* and *observed payoffs*. Duong et al. (2012) learns the structure of games that are *not graphical*, i.e., the payoff depends on all other players. Their approach also assumes *observed* payoff and consider a *dynamic* consensus scenario, where agents on a network attempt to reach a unanimous vote. In analogy to voting, we do not assume the availability of the dynamics (i.e., the previous actions) that led to the *final* vote. They also use fixed information on the conditioning sets of neighbors during their search for graph structure. We also note that the work of Vorobeychik et al. (2007); Gao and Pfeffer (2010); Ziebart et al. (2010) present experimental validation mostly for 2 players only, 7 players in Waugh et al. (2011) and up to 13 players in Duong et al. (2009).

In several cases in previous work, researchers define probabilistic models using knowledge of the payoff functions explicitly (i.e., a *Gibbs distribution* with potentials that are functions of the players payoffs, regrets, etc.) to model joint behavior (i.e., joint pure-strategies); see, e.g., Duong et al. (2008, 2010, 2012), and to some degree also Wright and Leyton-Brown (2010, 2012). It should be clear to the reader that this is not the same as our generative model, which is defined *directly* on the PSNE (or stable outcomes) of the game, which the players' payoffs determine only *indirectly*.

In contrast, in this paper, we assume that the joint actions are the *only* observable information and that both the game graph structure and payoff functions are *unknown, unobserved and unavailable*. We present the first techniques for learning the structure and parameters of a non-trivial class of large-population graphical games from joint actions only. Furthermore, we present experimental validation in games of up to 100 players. Our convex loss minimization approach could potentially be applied to larger problems since it has *polynomial time* complexity in the number of players.

2.1 On Learning Probabilistic Graphical Models

There has been a significant amount of work on learning the structure of *probabilistic* graphical models from data. We mention only a few references that follow a maximum likelihood approach for Markov random fields (Lee et al., 2007), bounded tree-width distributions (Chow and Liu, 1968; Srebro, 2001), Ising models (Wainwright et al., 2007; Banerjee et al., 2008; Höfling and Tibshirani, 2009), Gaussian graphical models (Banerjee et al., 2006), Bayesian networks (Guo and Schuurmans, 2006; Schmidt et al., 2007b) and directed cyclic graphs (Schmidt and Murphy, 2009).

Our approach learns the structure and parameters of games by maximum likelihood estimation on a related probabilistic model. Our probabilistic model does not fit into any of the types described above. Although a (directed) graphical game has a directed cyclic graph, there is a semantic difference with respect to graphical models. Structure in a graphical model implies a factorization of the probabilistic model. In a graphical game, the graph

structure implies *strategic* dependence between players, and has no immediate probabilistic implication. Furthermore, our general model differs from Schmidt and Murphy (2009) since our generative model does not decompose as a multiplication of potential functions.

Finally, it is very important to note that our specific aim is to model *behavioral data* that is *strategic* in nature. Hence, our modeling and learning approach deviates from those for *probabilistic* graphical models which are of course better suited for other types of data, mostly *probabilistic* in nature (i.e., resulting from a *fixed* underlying probability distribution). As a consequence, it is also very important to keep in mind that our work is not in competition with the work in probabilistic graphical models, and is not meant to replace it (except in the context of data sets collected from complex strategic behavior just mentioned). Each approach has its own aim, merits and pitfalls in terms of the nature of data sets that each seeks to model. We return to this point in Section 8 (Experimental Results).

2.2 On Linear Threshold Models and Econometrics

Irfan and Ortiz (2014) introduced LIGs in the AI community, showed that such games are useful, and addressed a variety of computational problems, including the identification of *most influential* senators. The class of LIGs is related to the well-known *linear threshold model (LTM)* in sociology (Granovetter, 1978), recently very popular within the social network and theoretical computer science community (Kleinberg, 2007).[9] Irfan and Ortiz (2014) discusses linear threshold models in depth; we briefly discuss them here for self-containment. LTMs are usually studied as the basis for some kind of diffusion process. A typical problem is the identification of most influential individuals in a social network. An LTM is not in itself a game-theoretic model and, in fact, Granovetter himself argues against this view in the context of the setting and the type of questions in which he was most interested (Granovetter, 1978). Our reading of the relevant literature suggests that subsequent work on LTMs has not taken a strictly game-theoretic view either. The problem of learning mathematical models of influence from behavioral data has just started to receive attention. There has been a number of articles in the last couple of years addressing the problem of learning the parameters of a variety of *diffusion* models of influence (Saito et al., 2008, 2009, 2010; Goyal et al., 2010; Gomez Rodriguez et al., 2010; Cao et al., 2011).[10]

Our model is also related to a particular model of *discrete choice with social interactions* in econometrics (see, e.g. Brock and Durlauf 2001). The main difference is that we take a strictly non-cooperative game-theoretic approach within the classical "static"/one-shot game framework and do not use a *random utility model*. We follow the approach of Irfan and Ortiz (2014) who takes a strictly non-cooperative game-theoretic approach within the classical "static"/one-shot game framework, and thus we do not use a *random utility model*. In addition, we do not make the assumption of *rational expectations*, which in the context

9. López-Pintado and Watts (2008) also provide an excellent summary of the various models in this area of mathematical social science.

10. Often learning consists of estimating the threshold parameter from data given as temporal sequences from "traces" or "action logs." Sometimes the "influence weights" are estimated assuming a given graph, and almost always the weights are assumed *positive* and estimated as "probabilities of influence." For example, Saito et al. (2010) considers a dynamic (continuous time) LTM that has only positive influence weights and a randomly generated threshold value. Cao et al. (2011) uses active learning to estimate the threshold values of an LTM leading to a maximum spread of influence.

of models of discrete choice with social interactions essentially implies the assumption that all players use *exactly the same mixed strategy.*[11]

3. Background: Game Theory and Linear Influence Games

In classical game-theory (see, e.g. Fudenberg and Tirole 1991 for a textbook introduction), a *normal-form game* is defined by a set of *players* V (e.g., we can let $V = \{1, \ldots, n\}$ if there are n players), and for each player i, a set of *actions*, or *pure-strategies* A_i, and a payoff function $u_i : \times_{j \in V} A_j \to \mathbb{R}$ mapping the joint actions of all the players, given by the Cartesian product $\mathcal{A} \equiv \times_{j \in V} A_j$, to a real number. In non-cooperative game theory we assume players are greedy, rational and act independently, by which we mean that each player i always want to maximize their own utility, subject to the actions selected by others, irrespective of how the optimal action chosen help or hurt others.

A core solution concept in non-cooperative game theory is that of an *Nash equilibrium.* A joint action $\mathbf{x}^* \in \mathcal{A}$ is a *pure-strategy Nash equilibrium (PSNE)* of a non-cooperative game if, for each player i, $x_i^* \in \arg\max_{x_i \in A_i} u_i(x_i, \mathbf{x}_{-i}^*)$; that is, $\mathbf{x}^*$ constitutes a *mutual best-response*, no player i has any incentive to unilaterally deviate from the prescribed action x_i^*, given the joint action of the other players $\mathbf{x}_{-i}^* \in \times_{j \in V - \{i\}} A_j$ in the equilibrium. In what follows, we denote a game by $\mathcal{G}$, and the set of all *pure-strategy Nash equilibria* of $\mathcal{G}$ by[12]

$$\mathcal{NE}(\mathcal{G}) \equiv \{\mathbf{x}^* \mid (\forall i \in V)\ x_i^* \in \arg\max_{x_i \in A_i} u_i(x_i, \mathbf{x}_{-i}^*)\}\ .$$

A *(directed) graphical game* is a game-theoretic graphical model (Kearns et al., 2001). It provides a succinct representation of normal-form games. In a graphical game, we have a (directed) graph $G = (V, E)$ in which each node in V corresponds to a player in the game. The interpretation of the edges/arcs E of G is that the payoff function of player i is only a function of the set of parents/neighbors $\mathcal{N}_i \equiv \{j \mid (i, j) \in E\}$ in G (i.e., the set of players corresponding to nodes that point to the node corresponding to player i in the graph). In the context of a graphical game, we refer to the u_i's as the *local payoff functions/matrices.*

Linear influence games (LIGs) (Irfan and Ortiz, 2014) are a sub-class of 2-action graphical games with *parametric* payoff functions. For LIGs, we assume that we are given a matrix of influence weights $\mathbf{W} \in \mathbb{R}^{n \times n}$, with $\mathbf{diag}(\mathbf{W}) = \mathbf{0}$, and a threshold vector $\mathbf{b} \in \mathbb{R}^n$. For each player i, we define the influence function $f_i(\mathbf{x}_{-i}) \equiv \sum_{j \in \mathcal{N}_i} w_{ij} x_j - b_i = \mathbf{w}_{i,-i}{}^{\mathrm{T}} \mathbf{x}_{-i} - b_i$ and the payoff function $u_i(\mathbf{x}) \equiv x_i f_i(\mathbf{x}_{-i})$. We further assume binary actions: $A_i \equiv \{-1, +1\}$ for all i. The *best response* x_i^* of player i to the joint action $\mathbf{x}_{-i}$ of the other players is defined as

$$\left.\begin{array}{l} \mathbf{w}_{i,-i}{}^{\mathrm{T}} \mathbf{x}_{-i} > b_i \Rightarrow x_i^* = +1, \\ \mathbf{w}_{i,-i}{}^{\mathrm{T}} \mathbf{x}_{-i} < b_i \Rightarrow x_i^* = -1 \text{ and} \\ \mathbf{w}_{i,-i}{}^{\mathrm{T}} \mathbf{x}_{-i} = b_i \Rightarrow x_i^* \in \{-1, +1\} \end{array}\right\} \Leftrightarrow x_i^* (\mathbf{w}_{i,-i}{}^{\mathrm{T}} \mathbf{x}_{-i} - b_i) \geq 0\ .$$

11. A formal definition of "rational expectations" is beyond the scope of this paper. We refer the reader to the early part of the article by Brock and Durlauf (2001) where they explain why assuming rational expectations leads to the conclusion that all players use exactly the same mixed strategy. That is the relevant part of that work to ours.
12. Because this paper concerns mostly PSNE, we denote the set of PSNE of game $\mathcal{G}$ as $\mathcal{NE}(\mathcal{G})$ to simplify notation.

Intuitively, for any other player j, we *can think of* $w_{ij} \in \mathbb{R}$ as a *weight* parameter quantifying the "influence factor" that j has on i, and we *can think of* $b_i \in \mathbb{R}$ as a *threshold* parameter quantifying the level of "tolerance" that player i has for playing -1.[13]

As discussed in Irfan and Ortiz (2014), LIGs are also a sub-class of polymatrix games (Janovskaja, 1968). Furthermore, in the special case of $\mathbf{b} = \mathbf{0}$ and symmetric $\mathbf{W}$, a LIG becomes a *party-affiliation game* (Fabrikant et al., 2004).

In this paper, the use of the verb "influence" strictly refers to influences defined by the model.

Figure 1 provides a preview illustration of the application of our approach to congressional voting.[14]

4. Our Proposed Framework for Learning LIGs

Our *goal* is to learn the structure and parameters of an LIG from observed joint actions only (i.e., without any payoff data/information).[15] Yet, for simplicity, most of the presentation in this section is actually in terms of general 2-action games. While we make sporadic references to LIGs throughout the section, it is not until we reach the end of the section that we present and discuss the particular instantiation of our proposed framework with LIGs.

Our main *performance measure* will be average log-likelihood (although later we will be considering misclassification-type error measures in the context of simultaneous-classification, as a result of an approximation of the average log-likelihood). Our *emphasis on a PSNE-*

13. As we *formally/mathematically define* here, LIGs are 2-action graphical games with *linear-quadratic* payoff functions. Given our main interest in this paper on the PSNE solution concept, for the most part, we simply view *LIGs as compact representations of the PSNE of graphical games that the algorithms of Irfan and Ortiz (2014) use for CSI.* (This is in contrast to a perhaps more natural, "intuitive" but still *informal* description/interpretation one may provide for instructive/pedagogical purposes based on "direct influences," as we do here.) This view of LIGs is analogous to the modern, predominant view of Bayesian networks as compact representations of joint probability distributions that are also very useful for modeling uncertainty in complex systems and practical for probabilistic inference (Koller and Friedman, 2009). (And also analogous is the "intuitive" descriptions/interpretations of BN structures, used for instructive/pedagogical purposes, based on "causal" interactions between the random variables Koller and Friedman, 2009.)

14. We present this game graph because many people express interest in "seeing" the type of games we learn on this particular data set. The reader should please understand that by presenting this graph we are *definitely not* implying or arguing that we can identify the ground-truth graph of "direct influences." (We say this even in the very unlikely event that the "ground-truth model" be an LIG that faithfully capture the "true direct influences" in this U.S. Congress, something arguably *no model* could ever do.) As we show later in Section 4.2, LIGs are *not* identifiable with respect to their *local* compact parametric representation encoding the game graph through their weights and biases, but only with respect to their PSNE, which are joint actions capturing a *global* property of a game that we really care about for CSI. Certainly, we could *never* validate the model parameters of an LIG at the *local*, microscopic level of "direct influences" using only the type of *observational* data we used to learn the model depicted by the graph in the figure. For that, we would need help from domain experts to design *controlled* experiments that would yield the right type of data for proper/rigorous scientific validation.

15. In principle, the *learning framework* itself is technically immediately/easily applicable to *any* class of simultaneous/one-shot games. Generalizing the algorithms and other theoretical results (e.g., on generalization error) while maintaining the tractability in sample complexity and computation may require significant effort.

based statistical model for the behavioral data results from the approach to causal strategic inference taken by Irfan and Ortiz (2014), which is strongly founded on PSNE.[16]

Note that our problem is *unsupervised*, i.e., we do not know a priori which joint actions are PSNE and which ones are not. If our only goal were to find a game $\mathcal{G}$ in which all the given observed data is an equilibrium, then any "dummy" game, such as the "dummy" LIG $\mathcal{G} = (\mathbf{W}, \mathbf{b}), \mathbf{W} = \mathbf{0}, \mathbf{b} = \mathbf{0}$, would be an optimal solution because $|\mathcal{NE}(\mathcal{G})| = 2^n$.[17] In this section, we present a probabilistic formulation that allows finding games that maximize the *empirical proportion of equilibria* in the data while keeping the *true proportion of equilibria* as low as possible. Furthermore, we show that *trivial* games such as LIGs with $\mathbf{W} = \mathbf{0}, \mathbf{b} = \mathbf{0}$, obtain the lowest log-likelihood.

4.1 Our Proposed Generative Model of Behavioral Data

We propose the following simple generative (mixture) model for behavioral data based strictly in the context of "simultaneous"/one-shot play in non-cooperative game theory, again motivated by Irfan and Ortiz (2014)'s PSNE-based approach to *causal strategic inference (CSI)*.[18] Let $\mathcal{G}$ be a game. With some probability $0 < q < 1$, a joint action $\mathbf{x}$ is chosen uniformly at random from $\mathcal{NE}(\mathcal{G})$; otherwise, $\mathbf{x}$ is chosen uniformly at random from its complement set $\{-1, +1\}^n - \mathcal{NE}(\mathcal{G})$. Hence, the generative model is a mixture model with mixture parameter q corresponding to the probability that a stable outcome (i.e., a PSNE) of the game is observed, uniform over PSNE. Formally, the probability mass function (PMF) over joint-behaviors $\{-1, +1\}^n$ parameterized by $(\mathcal{G}, q)$ is

$$p_{(\mathcal{G},q)}(\mathbf{x}) = q \frac{1[\mathbf{x} \in \mathcal{NE}(\mathcal{G})]}{|\mathcal{NE}(\mathcal{G})|} + (1 - q) \frac{1[\mathbf{x} \notin \mathcal{NE}(\mathcal{G})]}{2^n - |\mathcal{NE}(\mathcal{G})|}, \tag{1}$$

where we can think of q as the "signal" level, and thus $1 - q$ as the "noise" level in the data set.

Remark 1 *Note that in order for Eq.* (1) *to be a valid PMF for any $\mathcal{G}$, we need to enforce the following conditions $|\mathcal{NE}(\mathcal{G})| = 0 \Rightarrow q = 0$ and $|\mathcal{NE}(\mathcal{G})| = 2^n \Rightarrow q = 1$. Furthermore, note that in both cases ($|\mathcal{NE}(\mathcal{G})| \in \{0, 2^n\}$) the PMF becomes a uniform distribution. We also enforce the following condition:*[19] *if $0 < |\mathcal{NE}(\mathcal{G})| < 2^n$ then $q \notin \{0, 1\}$.*

16. The possibility that PSNE may not exist in some LIGs does not present a significant problem in our case because *we are learning the game, and can require that the LIG output has at least one PSNE.* Indeed, in our approach, games with no PSNE achieve the lowest possible likelihood within our generative model of the data; said differently, games with PSNE have higher likelihoods than those that do not have any PSNE.
17. Ng and Russell (2000) made a similar observation in the context of single-agent *inverse reinforcement learning (IRL)*.
18. Model "simplicity" and "abstractions" are not necessarily a bad thing in practice. More "realism" often leads to more "complexity" in terms of model representation and computation; and to potentially poorer generalization performance as well (Kearns and Vazirani, 1994). We believe that even if the data could be the result of complex cognitive, behavioral or neuronal processes underlying human decision making and social interactions, the practical guiding principle of model selection in ML, which governs the fundamental tradeoff between model complexity and generalization performance, still applies.
19. We can easily remove this condition at the expense of complicating the theoretical analysis on the generalization bounds because of having to deal with those extreme cases.

4.2 On PSNE-Equivalence and PSNE-Identifiability of Games

For any valid value of mixture parameter q, the PSNE of a game $\mathcal{G}$ completely determines our generative model $p_{(\mathcal{G},q)}$. Thus, given any such mixture parameter, two games with the same set of PSNE will induce the same PMF over the space of joint actions.[20]

Definition 2 *We say that two games $\mathcal{G}_1$ and $\mathcal{G}_2$ are* PSNE-equivalent *if and only if their PSNE sets are identical, i.e., $\mathcal{G}_1 \equiv_{\mathcal{NE}} \mathcal{G}_2 \Leftrightarrow \mathcal{NE}(\mathcal{G}_1) = \mathcal{NE}(\mathcal{G}_2)$.*

We often drop the "PSNE-" qualifier when clear from context.

Definition 3 *We say a set Υ^* of valid parameters $(\mathcal{G}, q)$ for the generative model is* PSNE-identifiable with respect to the PMF $p_{(\mathcal{G},q)}$ defined in Eq. (1)*, if and only if, for every pair $(\mathcal{G}_1, q_1), (\mathcal{G}_2, q_2) \in \Upsilon^*$, if $p_{(\mathcal{G}_1,q_1)}(\mathbf{x}) = p_{(\mathcal{G}_2,q_2)}(\mathbf{x})$ for all $\mathbf{x} \in \{-1,+1\}^n$ then $\mathcal{G}_1 \equiv_{\mathcal{NE}} \mathcal{G}_2$ and $q_1 = q_2$. We say a game $\mathcal{G}$ is* PSNE-identifiable with respect to Υ^* and the $p_{(\mathcal{G},q)}$*, if and only if, there exists a q such that $(\mathcal{G}, q) \in \Upsilon^*$.*

Definition 4 *We define the* true proportion of equilibria *in the game $\mathcal{G}$ relative to all possible joint actions as*

$$\pi(\mathcal{G}) \equiv |\mathcal{NE}(\mathcal{G})|/2^n \ . \tag{2}$$

We also say that a game $\mathcal{G}$ is trivial *if and only if $|\mathcal{NE}(\mathcal{G})| \in \{0, 2^n\}$ (or equivalently $\pi(\mathcal{G}) \in \{0, 1\}$), and* non-trivial *if and only if $0 < |\mathcal{NE}(\mathcal{G})| < 2^n$ (or equivalently $0 < \pi(\mathcal{G}) < 1$).*

The following propositions establish that the condition $q > \pi(\mathcal{G})$ ensures that the probability of an equilibrium is strictly greater than a non-equilibrium. The condition also guarantees that non-trivial games are identifiable.

Proposition 5 *Given a non-trivial game $\mathcal{G}$, the mixture parameter $q > \pi(\mathcal{G})$ if and only if $p_{(\mathcal{G},q)}(\mathbf{x}_1) > p_{(\mathcal{G},q)}(\mathbf{x}_2)$ for any $\mathbf{x}_1 \in \mathcal{NE}(\mathcal{G})$ and $\mathbf{x}_2 \notin \mathcal{NE}(\mathcal{G})$.*

Proof Note that $p_{(\mathcal{G},q)}(\mathbf{x}_1) = q/|\mathcal{NE}(\mathcal{G})| > p_{(\mathcal{G},q)}(\mathbf{x}_2) = (1-q)/(2^n - |\mathcal{NE}(\mathcal{G})|) \Leftrightarrow q > |\mathcal{NE}(\mathcal{G})|/2^n$ and given Eq. (2), we prove our claim. ■

Proposition 6 *Let $(\mathcal{G}_1, q_1)$ and $(\mathcal{G}_2, q_2)$ be two valid generative-model parameter tuples.*

(a) If $\mathcal{G}_1 \equiv_{\mathcal{NE}} \mathcal{G}_2$ and $q_1 = q_2$ then $(\forall \mathbf{x})\ p_{(\mathcal{G}_1,q_1)}(\mathbf{x}) = p_{(\mathcal{G}_2,q_2)}(\mathbf{x})$,

(b) Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be also two non-trivial games such that $q_1 > \pi(\mathcal{G}_1)$ and $q_2 > \pi(\mathcal{G}_2)$. If $(\forall \mathbf{x})\ p_{(\mathcal{G}_1,q_1)}(\mathbf{x}) = p_{(\mathcal{G}_2,q_2)}(\mathbf{x})$, then $\mathcal{G}_1 \equiv_{\mathcal{NE}} \mathcal{G}_2$ and $q_1 = q_2$.

20. It is not hard to come up with examples of multiple games that have the same PSNE set. In fact, later in this section, we show three instances of LIGs with very different weight-matrix parameter that have this property. Note that this is not a roadblock to our objectives of learning LIGs because our main interest is the PSNE of the game, not the individual parameters that define it. We note that this situation is hardly exclusive to game-theoretic models: an analogous issue occurs in probabilistic graphical models (e.g., Bayesian networks).

Proof Let $\mathcal{NE}_k \equiv \mathcal{NE}(\mathcal{G}_k)$. First, we prove part (a). By Definition 2, $\mathcal{G}_1 \equiv_{\mathcal{NE}} \mathcal{G}_2 \Rightarrow \mathcal{NE}_1 = \mathcal{NE}_2$. Note that $p_{(\mathcal{G}_k,q_k)}(\mathbf{x})$ in Eq. (1) depends only on characteristic functions $1[\mathbf{x} \in \mathcal{NE}_k]$. Therefore, $(\forall \mathbf{x})\ p_{(\mathcal{G}_1,q_1)}(\mathbf{x}) = p_{(\mathcal{G}_2,q_2)}(\mathbf{x})$.

Second, we prove part (b) by contradiction. Assume, wlog, $(\exists \mathbf{x})\ \mathbf{x} \in \mathcal{NE}_1 \wedge \mathbf{x} \notin \mathcal{NE}_2$. Then $p_{(\mathcal{G}_1,q_1)}(\mathbf{x}) = p_{(\mathcal{G}_2,q_2)}(\mathbf{x})$ implies by Eq. (1) that $q_1/|\mathcal{NE}_1| = (1-q_2)/(2^n - |\mathcal{NE}_2|) \Rightarrow q_1/\pi(\mathcal{G}_1) = (1-q_2)/(1-\pi(\mathcal{G}_2))$ by Eq. (2). By assumption, we have $q_1 > \pi(\mathcal{G}_1)$, which implies that $(1-q_2)/(1-\pi(\mathcal{G}_2)) > 1 \Rightarrow q_2 < \pi(\mathcal{G}_2)$, a contradiction. Hence, we have $\mathcal{G}_1 \equiv_{\mathcal{NE}} \mathcal{G}_2$. Assume, $q_1 \neq q_2$. Then we have $p_{(\mathcal{G}_1,q_1)}(\mathbf{x}) = p_{(\mathcal{G}_2,q_2)}(\mathbf{x})$ implies by Eq. (1) and $\mathcal{G}_1 \equiv_{\mathcal{NE}} \mathcal{G}_2$ (and after some algebraic manipulations) that $(q_1 - q_2)\left(\frac{1[\mathbf{x}\in\mathcal{NE}(\mathcal{G}_1)]}{|\mathcal{NE}(\mathcal{G}_1)|} - \frac{1[\mathbf{x}\notin\mathcal{NE}(\mathcal{G}_1)]}{2^n - |\mathcal{NE}(\mathcal{G}_1)|}\right) = 0 \Rightarrow \frac{1[\mathbf{x}\in\mathcal{NE}(\mathcal{G}_1)]}{|\mathcal{NE}(\mathcal{G}_1)|} = \frac{1[\mathbf{x}\notin\mathcal{NE}(\mathcal{G}_1)]}{2^n - |\mathcal{NE}(\mathcal{G}_1)|}$, a contradiction. ■

The last proposition, along with our definitions of "trivial" (as given in Definition 4) and "identifiable" (Definition 3), allows us to formally define our *hypothesis space*.

Definition 7 *Let $\mathcal{H}$ be a class of games of interest. We call the set $\Upsilon \equiv \{(\mathcal{G}, q) \mid \mathcal{G} \in \mathcal{H} \wedge 0 < \pi(\mathcal{G}) < q < 1\}$ the* hypothesis space of non-trivial identifiable games and mixture parameters. *We also refer to a game $\mathcal{G} \in \mathcal{H}$ that is also in some tuple $(\mathcal{G}, q) \in \Upsilon$ for some q, as a* non-trivial identifiable game.[21]

Remark 8 *Recall that a trivial game induces a uniform PMF by Remark 1. Therefore, a non-trivial game is not* equivalent *to a trivial game since by Proposition 5, non-trivial games do not induce uniform PMFs.*[22]

4.3 Additional Discussion on Modeling Choices

We now discuss other equilibrium concepts, such as mixed-strategy Nash equilibria (MSNE) and quantal response equilibria (QRE). We also discuss a more sophisticated noise process as well as a generalization of our model to non-uniform distributions; while likely more realistic, the alternative models are mathematically more complex and potentially less tractable computationally.

4.3.1 On Other Equilibrium Concepts

There is still quite a bit of debate as to the appropriateness of game-theoretic equilibrium concepts to model individual human behavior in a social context. Camerer's book on behavioral game theory (Camerer, 2003) addresses some of the issues. Our interpretation

21. Technically, we should call the set Υ "the hypothesis space consisting of tuples of non-trivial games from $\mathcal{H}$ and mixture parameters identifiable up to PSNE, with respect to the probabilistic model defined in Eq. (1)." Similarly, we should call such game $\mathcal{G}$ "a non-trivial game from $\mathcal{H}$ identifiable up to PSNE, with respect to the probabilistic model defined in Eq. (1)." We opted for brevity.

22. In general, Section 4.2 characterizes our hypothesis space (non-trivial identifiable games and mixture parameters) via two specific conditions. The first condition, *non-triviality* (explained in Remark 1), is $0 < \pi(\mathcal{G}) < 1$. The second condition, *identifiability of the PSNE set from its related PMF* (discussed in Propositions 5 and 6), is $\pi(G) < q$. For completeness, in this remark, we clarify that the class of trivial games (uniform PMFs) is different from the class of non-trivial games (non-uniform PMFs). Thus, in the rest of the paper we focus exclusively on non-trivial identifiable games; that is, games that produce non-uniform PMFs and for which the PSNE set is uniquely identified from their PMFs.

of Camerer's position is not that Nash equilibria is universally a bad predictor but that it is not *consistently* the best, for reasons that are still not well understood. This point is best illustrated in Chapter 3, Figure 3.1 of Camerer (2003).

(Logit) quantal response equilibria (QRE) (McKelvey and Palfrey, 1995) has been proposed as an alternative to Nash in the context of behavioral game theory. Models based on QRE have been shown superior during *initial play* in some experimental settings, but prior work assumes known structure and observed payoffs, and only the "precision/rationality parameter" is estimated, e.g. Wright and Leyton-Brown (2010, 2012). In a logit QRE, the precision parameter, typically denoted by λ, can be mathematically interpreted as the inverse-temperature parameter of individual Gibbs distributions over the pure-strategy of each player i.

It is customary to compute the MLE for λ from available data. To the best of our knowledge, all work in QRE assumes exact knowledge of the game payoffs, and thus, no method has been proposed to simultaneously estimate the payoff functions u_i when they are unknown. Indeed, computing MLE for λ, *given the payoff functions*, is relatively efficient for normal-form games using standard techniques, but may be hard for graphical games; on the other hand, MLE estimation of the payoff functions themselves within a QRE model of behavior seems like a highly non-trivial optimization problem, and is unclear that it is even computationally tractable, even in normal-form games. At the very least, standard techniques do not apply and more sophisticated optimization algorithms or heuristics would have to be derived. Such extensions are clearly beyond the scope of this paper.[23]

Wright and Leyton-Brown (2012) also considers even more mathematically complex variants of behavioral models that combine QRE with different models that account for constraints in "cognitive levels" of reasoning ability/effort, yet the estimation and usage of such models still assumes knowledge of the payoff functions.

It would be fair to say that most of the human-subject experiments in behavioral game theory involve only a handful of players (Camerer, 2003). The scalability of those results to games with a large population of players is unclear.

Now, just as an example, we do not necessarily view the Senators final votes as those of a *single* human individual anyway: after all, such a decision is (or should be) obtained with consultation with their staff and (one would at least hope) the constituency of the state they represent. Also, the final voting decision is taken after consultation or meetings between the staff of the different senators. We view this underlying process as one of "cheap talk." While cheap talk may be an important process to study, in this paper, we just concentrate on the end result: the final vote. The reason is more than just scientific; as the congressional voting setting illustrates, data for such process is sometimes not available, or would seem very hard to infer from the end-states alone. While our experiments concentrate on congressional voting data, because it is publicly available and easy to obtain, the same would hold for other settings such as Supreme court decisions, voluntary vaccination, UN voting records and governmental decisions, to name a few. We speculate that in almost

23. Note that despite the apparent similarity in mathematical expression between logit QRE and the PSNE of the LIG we obtain by using individual logistic regression, they are fundamentally different because of the complex correlations that QRE conditions impose on the parameters $(\mathbf{W}, \mathbf{b})$ of the payoff functions. It is unclear how to adapt techniques for logistic regression similar to the ones we used here to efficiently/tractably compute MLE within the logit QRE framework.

all those cases, only the end-result is likely to be recorded and little information would be available about the "cheap talk" process or "pre-play" period leading to the final decision.

In our work we consider PSNE because of our motivation to provide LIGs for use within the casual strategic inference framework for modeling "influence" in large-population networks of Irfan and Ortiz (2014). Note that the universality of MSNE does not diminish the importance of PSNE in game theory.[24] Indeed, a debate still exist within the game theory community as to the justification for randomization, specially in human contexts. While concentrating exclusively on PSNE may not make sense in *all* settings, it does make sense in *some*.[25] In addition, were we to introduce mixed-strategies into the inference and learning framework and model, we would be adding a considerable amount of complexity in almost all respects, thus requiring a substantive effort to study on its own.[26]

4.3.2 On the Noise Process

Here we discuss a more sophisticated noise process as well as a generalization of our model to non-uniform distributions. The problem with these models is that they lead to a significantly more complex expression for the generative model and thus likelihood functions. This is in contrast to the simplicity afforded us by the generative model with a more global noise process defined above. (See Appendix A.2.1 for further discussion.)

In this paper we considered a "global" noise process, modeled using a parameter q corresponding to the probability that a sample observation is an equilibrium of the underlying hidden game. One could easily envision potentially better and more natural/realistic "local" noise processes, at the expense of producing a significantly more complex generative model, and less computationally amenable, than the one considered in this paper. For instance, we could use a noise process that is formed of many independent, individual noise processes, one for each player. (See Appendix A.2.2 for further discussion.)

4.4 Learning Games via MLE

We now formulate the problem of learning games as one of maximum likelihood estimation with respect to our PSNE-based generative model defined in Eq. (1) and the hypothesis space of non-trivial identifiable games and mixture parameters (Definition 7). We remind

24. Research work on the properties and computation of PSNE include Rosenthal (1973); Gilboa and Zemel (1989); Stanford (1995); Rinott and Scarsini (2000); Fabrikant et al. (2004); Gottlob et al. (2005); Sureka and Wurman (2005); Daskalakis and Papadimitriou (2006); Dunkel (2007); Dunkel and Schulz (2006); Dilkina et al. (2007); Ackermann and Skopalik (2007); Hasan et al. (2008); Hasan and Galiana (2008); Ryan et al. (2010); Chapman et al. (2010); Hasan and Galiana (2010).

25. For example, in the context of congressional voting, we believe Senators almost always have full-information about how some, if not all other Senators they care about would vote. Said differently, we believe *uncertainty* in a Senator's final vote, *by the time the vote is actually taken*, is rare, and certainly not the norm. Hence, it is unclear how much there is to gain, *in this particular setting*, by considering possible randomization in the Senators' voting strategies.

26. For example, note that because in our setting we learn exclusively from observed joint actions, we could not assume knowledge of the internal mixed-strategies of players. Perhaps we could generalize our model to allow for mixed-strategies by defining a process in which a joint mixed strategy $\mathbf{p}$ from the set of MSNE (or its complement) is drawn according to some distribution, then a (pure-strategy) realization $\mathbf{x}$ is drawn from $\mathbf{p}$ that would correspond to the observed joint actions. One problem we might face with this approach is that little is known about the structure of MSNE in general multi-player games. For example, it is not even clear that the set of MSNE is always measurable in general!

the reader that our problem is unsupervised; that is, we do not know a priori which joint actions are equilibria and which ones are not. We base our framework on the fact that *games are PSNE-identifiable* with respect to their induced PMF, under the condition that $q > \pi(\mathcal{G})$, by Proposition 6.

First, we introduce a shorthand notation for the *Kullback-Leibler (KL) divergence* between two Bernoulli distributions parameterized by $0 \leq p_1 \leq 1$ and $0 \leq p_2 \leq 1$:

$$\begin{aligned} KL(p_1 \| p_2) &\equiv KL(\text{Bernoulli}(p_1) \| \text{Bernoulli}(p_2)) \\ &= p_1 \log \tfrac{p_1}{p_2} + (1-p_1) \log \tfrac{1-p_1}{1-p_2} \ . \end{aligned} \tag{3}$$

Using this function, we can derive the following expression of the MLE problem.

Lemma 9 *Given a data set $\mathcal{D} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}\}$, define the* empirical proportion of equilibria*, i.e., the proportion of samples in $\mathcal{D}$ that are equilibria of $\mathcal{G}$, as*

$$\widehat{\pi}(\mathcal{G}) \equiv \tfrac{1}{m} \textstyle\sum_l 1[\mathbf{x}^{(l)} \in \mathcal{NE}(\mathcal{G})] \ . \tag{4}$$

The MLE problem for the probabilistic model given in Eq. (1) *can be expressed as finding:*

$$(\widehat{\mathcal{G}}, \widehat{q}) \in \arg\max_{(\mathcal{G},q)\in\Upsilon} \widehat{\mathcal{L}}(\mathcal{G}, q), \text{where } \widehat{\mathcal{L}}(\mathcal{G}, q) = KL(\widehat{\pi}(\mathcal{G}) \| \pi(\mathcal{G})) - KL(\widehat{\pi}(\mathcal{G}) \| q) - n \log 2 \ , \tag{5}$$

where $\mathcal{H}$ and Υ are as in Definition 7, and $\pi(\mathcal{G})$ is defined as in Eq. (2)*. Also, the optimal mixture parameter $\widehat{q} = \min(\widehat{\pi}(\mathcal{G}), 1 - \frac{1}{2m})$.*

Proof Let $\mathcal{NE} \equiv \mathcal{NE}(\mathcal{G})$, $\pi \equiv \pi(\mathcal{G})$ and $\widehat{\pi} \equiv \widehat{\pi}(\mathcal{G})$. First, for a non-trivial $\mathcal{G}$, $\log p_{(\mathcal{G},q)}(\mathbf{x}^{(l)}) = \log \frac{q}{|\mathcal{NE}|}$ for $\mathbf{x}^{(l)} \in \mathcal{NE}$, and $\log p_{(\mathcal{G},q)}(\mathbf{x}^{(l)}) = \log \frac{1-q}{2^n - |\mathcal{NE}|}$ for $\mathbf{x}^{(l)} \notin \mathcal{NE}$. The average log-likelihood $\widehat{\mathcal{L}}(\mathcal{G}, q) = \frac{1}{m} \sum_l \log p_{\mathcal{G},q}(\mathbf{x}^{(l)}) = \widehat{\pi} \log \frac{q}{|\mathcal{NE}|} + (1-\widehat{\pi}) \log \frac{1-q}{2^n - |\mathcal{NE}|} = \widehat{\pi} \log \frac{q}{\pi} + (1-\widehat{\pi}) \log \frac{1-q}{1-\pi} - n \log 2$. By adding $0 = -\widehat{\pi} \log \widehat{\pi} + \widehat{\pi} \log \widehat{\pi} - (1-\widehat{\pi}) \log(1-\widehat{\pi}) + (1-\widehat{\pi}) \log(1-\widehat{\pi})$, this can be rewritten as $\widehat{\mathcal{L}}(\mathcal{G}, q) = \widehat{\pi} \log \frac{\widehat{\pi}}{\pi} + (1-\widehat{\pi}) \log \frac{1-\widehat{\pi}}{1-\pi} - \widehat{\pi} \log \frac{\widehat{\pi}}{q} - (1-\widehat{\pi}) \log \frac{1-\widehat{\pi}}{1-q} - n \log 2$, and by using Eq. (3) we prove our claim.

Note that by maximizing with respect to the mixture parameter q and by properties of the KL divergence, we get $KL(\widehat{\pi} \| \widehat{q}) = 0 \Leftrightarrow \widehat{q} = \widehat{\pi}$. We define our hypothesis space Υ given the conditions in Remark 1 and Propositions 5 and 6. For the case $\widehat{\pi} = 1$, we "shrink" the optimal mixture parameter $\widehat{q}$ to $1 - \frac{1}{2m}$ in order to enforce the second condition given in Remark 1. ■

Remark 10 *Recall that a trivial game (e.g., LIG $\mathcal{G} = (\mathbf{W}, \mathbf{b}), \mathbf{W} = \mathbf{0}, \mathbf{b} = \mathbf{0}, \pi(\mathcal{G}) = 1$) induces a uniform PMF by Remark 1, and therefore its log-likelihood is $-n \log 2$. Note that the lowest log-likelihood for non-trivial identifiable games in Eq.* (5) *is $-n \log 2$ by setting the optimal mixture parameter $\widehat{q} = \widehat{\pi}(\mathcal{G})$ and given that $KL(\widehat{\pi}(\mathcal{G}) \| \pi(\mathcal{G})) \geq 0$.*

Furthermore, Eq. (5) implies that for non-trivial identifiable games $\mathcal{G}$, we expect the *true proportion of equilibria* $\pi(\mathcal{G})$ to be strictly less than the *empirical proportion of equilibria* $\widehat{\pi}(\mathcal{G})$ in the given data. This is by setting the optimal mixture parameter $\widehat{q} = \widehat{\pi}(\mathcal{G})$ and the condition $q > \pi(\mathcal{G})$ in our hypothesis space.

4.4.1 Learning LIGs via MLE: Model Selection

Our main *learning problem* consists of inferring the structure and parameters of an LIG from data *with the main purpose being modeling the game's PSNE, as reflected in the generative model.* Note that, as we have previously stated, different games (i.e., with different payoff functions) can be PSNE-equivalent. For instance, the three following LIGs, with different weight parameter matrices, induce the same PSNE sets, i.e., $\mathcal{NE}(\mathbf{W}_k, \mathbf{0}) = \{(-1,-1,-1),(+1,+1,+1)\}$ for $k = 1, 2, 3$:[27]

$$\mathbf{W}_1 = \begin{bmatrix} 0 & 0 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad \mathbf{W}_2 = \begin{bmatrix} 0 & 0 & 0 \\ 2 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad \mathbf{W}_3 = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}.$$

Thus, not only the MLE may not be unique, but also all such PSNE-equivalent MLE games will achieve the same level of *generalization* performance. But, as reflected by our generative model, our main interest in the model parameters of the LIGs is only with respect to the PSNE they induce, not the model parameters *per se*. Hence, all we need is a way to select among PSNE-equivalent LIGs.

In our work, *the indentifiability or interpretability of exact model parameters of LIGs is not our main interest.* That is, in the research presented here, we did not seek or attempt to work on creating alternative generative models with the objective to provide a theoretical guarantee that, given an infinite amount of data, we can recover the model parameters of an unknown ground-truth model generating the data, assuming the ground-truth model is an LIG. We opted for a more practical ML approach in which we just want to learn a *single* LIG that has good generalization performance (i.e., predictive performance in terms of average log-likelihood) with respect to our generative model. Given the nature of our generative model, this essentially translate to learning an LIG that captures as best as possible the PSNE of the unknown ground-truth game. Unfortunately, as we just illustrated, several LIGs with different model parameter values can have the same set of PSNE. Thus, they all would have the same (generalization) performance ability.

As we all know, model selection is core to ML. One of the reason we chose an ML-approach to learning games is precisely the elegant way in which ML deals with the problem of how to select among multiple models that achieve the same level of performance: invoke the principle of Ockham's razor and select the "simplest" model among those with the same (generalization) performance. This ML philosophy is not *ad hoc*. It is instead well established in practice and well supported by theory. Seminal results from the various theories of learning, such as computational and statistical learning theory and PAC learning, support the well-known ML adage that "learning requires bias." In short, as is by now

27. Using the formal mathematical definition of "identifiability" in statistics, we would say that the LIG examples prove that the model parameters $(\mathbf{W}, \mathbf{b})$ of an LIG $\mathcal{G}$ are *not* identifiable with respect to the generative model $p_{(\mathcal{G},q)}$ defined in Eq. (1). We note that this situation is hardly exclusive to game-theoretic models. As example of an analogous issue in probabilistic graphical models is the fact that two Bayesian networks with different graph structures can represent not only the same conditional independence properties but also *exactly* the same set of joint probability distributions (Chickering, 2002; Koller and Friedman, 2009).

    As a side note, we can distinguish these games with respect to their larger set of mixed-strategy Nash equilibria (MSNE), but, as stated previously, we do not consider MSNE in this paper because our primary motivation is the work of Irfan and Ortiz (2014), which is based on the concept of PSNE.

standard in an ML-approach, we measure the quality of our data-induced models via their generalization ability and invoke the principle of Ockham's razor to bias our search toward simpler models using well-known and -studied regularization techniques.

Now, as we also all know, exactly what "simple" and "bias" means depends on the problem. In our case, we would prefer games with sparse graphs, if for no reason other than to simplify analysis, exploration, study, and (*visual*) "interpretability" of the game model *by human experts*, everything else being equal (i.e., models with the same explanatory power on the data as measured by the likelihoods).[28] For example, among the LIGs presented above, using structural properties alone, we would generally prefer the former two models to the latter, all else being equal (e.g., generalization performance).

5. Generalization Bound and VC-Dimension

In this section, we show a generalization bound for the maximum likelihood problem as well as an upper bound of the VC-dimension of LIGs. Our objective is to establish that with probability at least $1-\delta$, for some confidence parameter $0 < \delta < 1$, the maximum likelihood estimate is within $\epsilon > 0$ of the optimal parameters, in terms of achievable expected log-likelihood.

Given the ground-truth distribution $\mathcal{Q}$ of the data, let $\bar{\pi}(\mathcal{G})$ be the *expected proportion of equilibria*, i.e.,

$$\bar{\pi}(\mathcal{G}) = \mathbb{P}_{\mathcal{Q}}[\mathbf{x} \in \mathcal{NE}(\mathcal{G})] \ ,$$

and let $\bar{\mathcal{L}}(\mathcal{G}, q)$ be the *expected log-likelihood* of a generative model from game $\mathcal{G}$ and mixture parameter q, i.e.,

$$\bar{\mathcal{L}}(\mathcal{G}, q) = \mathbb{E}_{\mathcal{Q}}[\log p_{(\mathcal{G},q)}(\mathbf{x})] \ .$$

Let $\widehat{\theta} \equiv (\widehat{\mathcal{G}}, \widehat{q})$ be a maximum-likelihood estimate as in Eq. (5) (i.e., $\widehat{\theta} \in \arg\max_{\theta \in \Upsilon} \widehat{\mathcal{L}}(\theta)$), and $\bar{\theta} \equiv (\bar{\mathcal{G}}, \bar{q})$ be the *maximum-expected-likelihood estimate*: $\bar{\theta} \in \arg\max_{\theta \in \Upsilon} \bar{\mathcal{L}}(\theta)$.[29] We use, without formally re-stating, the last definitions in the technical results presented in the remaining of this section.

Note that our hypothesis space Υ as stated in Definition 7 includes a continuous parameter q that could potentially have infinite VC-dimension. The following lemma will allow us later to prove that uniform convergence for the extreme values of q implies uniform convergence for all q in the domain.

28. Just to be clear, here we mean "interpretability" not in any formal mathematical sense, or as often used in some areas of the social sciences such as economics. But, instead, as we typically use it in ML/AI textbooks, such as for example, when referring to shallow/sparse decision trees, generally considered to be easier to explain and understand. Similarly, the hope here is that the "sparsity" or "simplicity" of the game graph/model would make it also simpler for human experts to explain or understand what about the model is leading them to generate novel hypotheses, reach certain conclusions or make certain inferences about the *global* strategic behavior of the agents/players, such as those based on the game's PSNE and facilitated by CSI. We should also point out that, in preliminary empirical work, we have observed that the *representationally sparser* the LIG graph, the *computationally easier* it is for algorithms and other heuristics that operate on the LIG, as those of Irfan and Ortiz (2014) for CSI, for example.

29. If the ground-truth model belongs to the class of LIGs, then $\bar{\theta}$ is also the ground-truth model, or PSNE-equivalent to it.

Lemma 11 *Consider any game $\mathcal{G}$ and, for $0 < q'' < q' < q < 1$, let $\theta = (\mathcal{G}, q)$, $\theta' = (\mathcal{G}, q')$ and $\theta'' = (\mathcal{G}, q'')$. If, for any $\epsilon > 0$ we have $|\widehat{\mathcal{L}}(\theta) - \bar{\mathcal{L}}(\theta)| \leq \epsilon/2$ and $|\widehat{\mathcal{L}}(\theta'') - \bar{\mathcal{L}}(\theta'')| \leq \epsilon/2$, then $|\widehat{\mathcal{L}}(\theta') - \bar{\mathcal{L}}(\theta')| \leq \epsilon/2$.*

Proof Let $\mathcal{NE} \equiv \mathcal{NE}(\mathcal{G})$, $\pi \equiv \pi(\mathcal{G})$, $\widehat{\pi} \equiv \widehat{\pi}(\mathcal{G})$, $\bar{\pi} \equiv \bar{\pi}(\mathcal{G})$, and $\mathbb{E}[\cdot]$ and $\mathbb{P}[\cdot]$ be the expectation and probability with respect to the ground-truth distribution $\mathcal{Q}$ of the data.

First note that for any $\theta = (\mathcal{G}, q)$, we have $\bar{\mathcal{L}}(\theta) = \mathbb{E}[\log p_{(\mathcal{G},q)}(\mathbf{x})] = \mathbb{E}[1[\mathbf{x} \in \mathcal{NE}] \log \frac{q}{|\mathcal{NE}|} + 1[\mathbf{x} \notin \mathcal{NE}] \log \frac{1-q}{2^n - |\mathcal{NE}|}] = \mathbb{P}[\mathbf{x} \in \mathcal{NE}] \log \frac{q}{|\mathcal{NE}|} + \mathbb{P}[\mathbf{x} \notin \mathcal{NE}] \log \frac{1-q}{2^n - |\mathcal{NE}|} = \bar{\pi} \log \frac{q}{|\mathcal{NE}|} + (1 - \bar{\pi}) \log \frac{1-q}{2^n - |\mathcal{NE}|} = \bar{\pi} \log \left(\frac{q}{1-q} \cdot \frac{2^n - |\mathcal{NE}|}{|\mathcal{NE}|} \right) + \log \frac{1-q}{2^n - |\mathcal{NE}|} = \bar{\pi} \log \left(\frac{q}{1-q} \cdot \frac{1-\pi}{\pi} \right) + \log \frac{1-q}{1-\pi} - n \log 2$.

Similarly, for any $\theta = (\mathcal{G}, q)$, we have $\widehat{\mathcal{L}}(\theta) = \widehat{\pi} \log \left(\frac{q}{1-q} \cdot \frac{1-\pi}{\pi} \right) + \log \frac{1-q}{1-\pi} - n \log 2$. So that $\widehat{\mathcal{L}}(\theta) - \bar{\mathcal{L}}(\theta) = (\widehat{\pi} - \bar{\pi}) \log \left(\frac{q}{1-q} \cdot \frac{1-\pi}{\pi} \right)$.

Furthermore, the function $\frac{q}{1-q}$ is strictly monotonically increasing for $0 \leq q < 1$. If $\widehat{\pi} > \bar{\pi}$ then $-\epsilon/2 \leq \widehat{\mathcal{L}}(\theta'') - \bar{\mathcal{L}}(\theta'') < \widehat{\mathcal{L}}(\theta') - \bar{\mathcal{L}}(\theta') < \widehat{\mathcal{L}}(\theta) - \bar{\mathcal{L}}(\theta) \leq \epsilon/2$. Else, if $\widehat{\pi} < \bar{\pi}$, we have $\epsilon/2 \geq \widehat{\mathcal{L}}(\theta'') - \bar{\mathcal{L}}(\theta'') > \widehat{\mathcal{L}}(\theta') - \bar{\mathcal{L}}(\theta') > \widehat{\mathcal{L}}(\theta) - \bar{\mathcal{L}}(\theta) \geq -\epsilon/2$. Finally, if $\widehat{\pi} = \bar{\pi}$ then $\widehat{\mathcal{L}}(\theta'') - \bar{\mathcal{L}}(\theta'') = \widehat{\mathcal{L}}(\theta') - \bar{\mathcal{L}}(\theta') = \widehat{\mathcal{L}}(\theta) - \bar{\mathcal{L}}(\theta) = 0$. ■

In the remaining of this section, denote by $d(\mathcal{H}) \equiv |\cup_{\mathcal{G} \in \mathcal{H}} \{\mathcal{NE}(\mathcal{G})\}|$ the number of all possible PSNE sets induced by games in $\mathcal{H}$, the class of games of interest.

The following theorem shows that the expected log-likelihood of the maximum likelihood estimate $\widehat{\theta}$ converges in probability to that of the optimal $\bar{\theta} = (\bar{\mathcal{G}}, \bar{q})$, as the data size m increases.

Theorem 12 *The following holds with $\mathcal{Q}$-probability at least $1 - \delta$:*

$$\bar{\mathcal{L}}(\widehat{\theta}) \geq \bar{\mathcal{L}}(\bar{\theta}) - \left(\log \max(2m, \tfrac{1}{1-\bar{q}}) + n \log 2 \right) \sqrt{\tfrac{2}{m} \left(\log d(\mathcal{H}) + \log \tfrac{4}{\delta} \right)} .$$

Proof First our objective is to find a lower bound for $\mathbb{P}[\bar{\mathcal{L}}(\widehat{\theta}) - \bar{\mathcal{L}}(\bar{\theta}) \geq -\epsilon] \geq \mathbb{P}[\bar{\mathcal{L}}(\widehat{\theta}) - \bar{\mathcal{L}}(\bar{\theta}) \geq -\epsilon + (\widehat{\mathcal{L}}(\widehat{\theta}) - \widehat{\mathcal{L}}(\bar{\theta}))] \geq \mathbb{P}[-\widehat{\mathcal{L}}(\widehat{\theta}) + \bar{\mathcal{L}}(\widehat{\theta}) \geq -\frac{\epsilon}{2}, \widehat{\mathcal{L}}(\bar{\theta}) - \bar{\mathcal{L}}(\bar{\theta}) \geq -\frac{\epsilon}{2}] = \mathbb{P}[\widehat{\mathcal{L}}(\widehat{\theta}) - \bar{\mathcal{L}}(\widehat{\theta}) \leq \frac{\epsilon}{2}, \widehat{\mathcal{L}}(\bar{\theta}) - \bar{\mathcal{L}}(\bar{\theta}) \geq -\frac{\epsilon}{2}] = 1 - \mathbb{P}[\widehat{\mathcal{L}}(\widehat{\theta}) - \bar{\mathcal{L}}(\widehat{\theta}) > \frac{\epsilon}{2} \vee \widehat{\mathcal{L}}(\bar{\theta}) - \bar{\mathcal{L}}(\bar{\theta}) < -\frac{\epsilon}{2}]$.

Let $\widetilde{q} \equiv \max(1 - \frac{1}{2m}, \bar{q})$. Now, we have $\mathbb{P}[\widehat{\mathcal{L}}(\widehat{\theta}) - \bar{\mathcal{L}}(\widehat{\theta}) > \frac{\epsilon}{2} \vee \widehat{\mathcal{L}}(\bar{\theta}) - \bar{\mathcal{L}}(\bar{\theta}) < -\frac{\epsilon}{2}] \leq \mathbb{P}[(\exists \theta \in \Upsilon, q \leq \widetilde{q}) \; |\widehat{\mathcal{L}}(\theta) - \bar{\mathcal{L}}(\theta)| > \frac{\epsilon}{2}] = \mathbb{P}[(\exists \theta, \mathcal{G} \in \mathcal{H}, q \in \{\pi(\mathcal{G}), \widetilde{q}\}) \; |\widehat{\mathcal{L}}(\theta) - \bar{\mathcal{L}}(\theta)| > \frac{\epsilon}{2}]$. The last equality follows from invoking Lemma 11.

Note that $\mathbb{E}[\widehat{\mathcal{L}}(\theta)] = \bar{\mathcal{L}}(\theta)$ and that since $\pi(\mathcal{G}) \leq q \leq \widetilde{q}$, the log-likelihood is bounded as $(\forall \mathbf{x}) \; -B \leq \log p_{(\mathcal{G},q)}(\mathbf{x}) \leq 0$, where $B = \log \frac{1}{1-\widetilde{q}} + n \log 2 = \log \max(2m, \frac{1}{1-\bar{q}}) + n \log 2$. Therefore, by Hoeffding's inequality, we have $\mathbb{P}[|\widehat{\mathcal{L}}(\theta) - \bar{\mathcal{L}}(\theta)| > \frac{\epsilon}{2}] \leq 2e^{-\frac{m\epsilon^2}{2B^2}}$.

Furthermore, note that there are $2d(\mathcal{H})$ possible parameters θ, since we need to consider only two values of $q \in \{\pi(\mathcal{G}), \widetilde{q}\}$) and because the number of all possible PSNE sets induced by games in $\mathcal{H}$ is $d(\mathcal{H}) \equiv |\cup_{\mathcal{G} \in \mathcal{H}} \{\mathcal{NE}(\mathcal{G})\}|$. Therefore, by the union bound we get the following uniform convergence $\mathbb{P}[(\exists \theta, \mathcal{G} \in \mathcal{H}, q \in \{\pi(\mathcal{G}), \widetilde{q}\}) \; |\widehat{\mathcal{L}}(\theta) - \bar{\mathcal{L}}(\theta)| > \frac{\epsilon}{2}] \leq 4d(\mathcal{H}) \mathbb{P}[|\widehat{\mathcal{L}}(\theta) - \bar{\mathcal{L}}(\theta)| > \frac{\epsilon}{2}] \leq 4d(\mathcal{H}) e^{-\frac{m\epsilon^2}{2B^2}} = \delta$. Finally, by solving for δ we prove our claim. ■

Remark 13 *A more elaborate analysis allows to tighten the bound in Theorem 12 from* $\mathcal{O}(\log \frac{1}{1-\bar{q}})$ *to* $\mathcal{O}(\log \frac{\bar{q}}{1-\bar{q}})$. *We chose to provide the former result for clarity of presentation.*

The following theorem establishes the complexity of the class of LIGs, which implies that the term $\log d(\mathcal{H})$ of the generalization bound in Theorem 12 is only polynomial in the number of players n.

Theorem 14 *If* $\mathcal{H}$ *is the class of* LIGs*, then* $d(\mathcal{H}) \equiv |\cup_{\mathcal{G}\in\mathcal{H}}\{\mathcal{NE}(\mathcal{G})\}| \leq 2^{n\frac{n(n+1)}{2}+1} \leq 2^{n^3}$.

Proof The logarithm of the number of possible pure-strategy Nash equilibria sets supported by $\mathcal{H}$ (i.e., that can be produced by some game in $\mathcal{H}$) is upper bounded by the VC-dimension of the class of neural networks with a single hidden layer of n units and $n + \binom{n}{2}$ input units, linear threshold activation functions, and constant output weights.

For every LIG $\mathcal{G} = (\mathbf{W}, \mathbf{b})$ in $\mathcal{H}$, define the neural network with a single layer of n hidden units, n of the inputs corresponds to the linear terms $x_1, \ldots, x_n$ and $\binom{n}{2}$ corresponds to the quadratic polynomial terms $x_i x_j$ for all pairs of players (i, j), $1 \leq i < j \leq n$. For every hidden unit i, the weights corresponding to the linear terms $x_1, \ldots, x_n$ are $-b_1, \ldots, -b_n$, respectively, while the weights corresponding to the quadratic terms $x_i x_j$ are $-w_{ij}$, for all pairs of players (i, j), $1 \leq i < j \leq n$, respectively. The weights of the bias term of all the hidden units are set to 0. All n output weights are set to 1 while the weight of the output bias term is set to 0. The output of the neural network is $1[\mathbf{x} \notin \mathcal{NE}(\mathcal{G})]$. Note that we define the neural network to classify non-equilibrium as opposed to equilibrium to keep the convention in the neural network literature to define the threshold function to output 0 for input 0. The alternative is to redefine the threshold function to output 1 instead for input 0.

Finally, we use the VC-dimension of neural networks (Sontag, 1998). ■

From Theorems 12 and 14, we state the generalization bounds for LIGs.

Corollary 15 *The following holds with* $\mathcal{Q}$*-probability at least* $1 - \delta$*:*

$$\bar{\mathcal{L}}(\widehat{\theta}) \geq \bar{\mathcal{L}}(\bar{\theta}) - \left(\log \max(2m, \tfrac{1}{1-\bar{q}}) + n \log 2\right) \sqrt{\tfrac{2}{m}\left(n^3 \log 2 + \log \tfrac{4}{\delta}\right)}\,,$$

where $\mathcal{H}$ *is the class of* LIGs*, in which case* $\Upsilon \equiv \{(\mathcal{G}, q) \mid \mathcal{G} \in \mathcal{H} \wedge 0 < \pi(\mathcal{G}) < q < 1\}$ *(Definition 7) becomes the hypothesis space of non-trivial identifiable* LIGs *and mixture parameters.*

6. Algorithms

In this section, we approximate the maximum likelihood problem by maximizing the number of observed equilibria in the data, suitable for a hypothesis space of games with small true proportion of equilibria. We then present our convex loss minimization approach. We also discuss baseline methods such as sigmoidal approximation and exhaustive search.

But first, let us discuss some negative results that justifies the use of simple approaches. Irfan and Ortiz (2014) showed that counting the number of Nash equilibria in LIGs is

Algorithm 1 Sample-Picking for General Games

Input: Data set $\mathcal{D} = \mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}$.
Compute the unique samples $\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(U)}$ and their frequency $\widehat{p}^{(1)}, \ldots, \widehat{p}^{(U)}$ in the data set $\mathcal{D}$.
Sort joint actions by their frequency such that $\widehat{p}^{(1)} \geq \widehat{p}^{(2)} \geq \cdots \geq \widehat{p}^{(U)}$.
for each unique sample $k = 1, \ldots, U$ **do**
  Define $\mathcal{G}_k$ by the Nash equilibria set $\mathcal{NE}(\mathcal{G}_k) = \{\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(k)}\}$.
  Compute the log-likelihood $\widehat{\mathcal{L}}(\mathcal{G}_k, \widehat{q}_k)$ in Eq. (5) (Note that $\widehat{q}_k = \widehat{\pi}(\mathcal{G}) = \frac{1}{m}(\widehat{p}^{(1)} + \cdots + \widehat{p}^{(k)})$, $\pi(\mathcal{G}) = \frac{k}{2^n}$).
end for
Output: The game $\mathcal{G}_{\widehat{k}}$ such that $\widehat{k} = \arg\max_k \widehat{\mathcal{L}}(\mathcal{G}_k, \widehat{q}_k)$.

#P-complete; thus, computing the log-likelihood function, and therefore MLE, is NP-hard.[30] General approximation techniques such as pseudo-likelihood estimation do not lead to tractable methods for learning LIGs.[31] From an optimization perspective, the log-likelihood function is not continuous because of the number of equilibria. Therefore, we cannot rely on concepts such as Lipschitz continuity.[32] Furthermore, bounding the number of equilibria by known bounds for Ising models leads to trivial bounds.[33]

6.1 An Exact Quasi-Linear Method for General Games: Sample-Picking

As a first approach, consider solving the maximum likelihood estimation problem in Eq. (5) by an exact exhaustive search algorithm. This algorithm iterates through all possible Nash equilibria sets, i.e., for $s = 0, \ldots, 2^n$, we generate all possible sets of size s with elements from the joint-action space $\{-1, +1\}^n$. Recall that there exist $\binom{2^n}{s}$ of such sets of size s and since $\sum_{s=0}^{2^n} \binom{2^n}{s} = 2^{2^n}$ the search space is super-exponential in the number of players n.

Based on few observations, we can obtain an $\mathcal{O}(m \log m)$ algorithm for m samples. First, note that the above method does not constrain the set of Nash equilibria in any fashion. Therefore, only joint actions that are observed in the data are candidates of being Nash equilibria in order to maximize the log-likelihood. This is because the introduction of an

30. This is not a disadvantage relative to probabilistic graphical models, since computing the log-likelihood function is also NP-hard for Ising models and Markov random fields in general, while learning is also NP-hard for Bayesian networks.

31. We show that evaluating the pseudo-likelihood function for our generative model is NP-hard. Consider a non-trivial LIG $\mathcal{G} = (\mathbf{W}, \mathbf{b})$. Furthermore, assume $\mathcal{G}$ has a single *non-absolutely-indifferent* player i and *absolutely-indifferent* players $\forall j \neq i$; that is, assume that $(\mathbf{w}_{i,-i}, b_i) \neq \mathbf{0}$ and $(\forall j \neq i)$ $(\mathbf{w}_{j,-j}, b_j) = \mathbf{0}$ (See Definition 19). Let $f_i(\mathbf{x}_{-i}) \equiv \mathbf{w}_{i,-i}{}^{\mathrm{T}} \mathbf{x}_{-i} - b_i$, we have $1[\mathbf{x} \in \mathcal{NE}(\mathcal{G})] = 1[x_i f_i(\mathbf{x}_{-i}) \geq 0]$ and therefore $p_{(\mathcal{G},q)}(\mathbf{x}) = q \frac{1[x_i f_i(\mathbf{x}_{-i}) \geq 0]}{|\mathcal{NE}(\mathcal{G})|} + (1-q) \frac{1 - 1[x_i f_i(\mathbf{x}_{-i}) \geq 0]}{2^n - |\mathcal{NE}(\mathcal{G})|}$. The result follows because computing $|\mathcal{NE}(\mathcal{G})|$ is #P-complete, even for this specific instance of a single *non-absolutely-indifferent* player (Irfan and Ortiz, 2014).

32. To prove that counting the number of equilibria is not (Lipschitz) continuous, we show how small changes in the parameters $\mathcal{G} = (\mathbf{W}, \mathbf{b})$ can produce big changes in $|\mathcal{NE}(\mathcal{G})|$. For instance, consider two games $\mathcal{G}_k = (\mathbf{W}_k, \mathbf{b}_k)$, where $\mathbf{W}_1 = \mathbf{0}, \mathbf{b}_1 = \mathbf{0}, |\mathcal{NE}(\mathcal{G}_1)| = 2^n$ and $\mathbf{W}_2 = \varepsilon(\mathbf{1}\mathbf{1}^{\mathrm{T}} - \mathbf{I}), \mathbf{b}_2 = \mathbf{0}, |\mathcal{NE}(\mathcal{G}_2)| = 2$ for $\varepsilon > 0$. For $\varepsilon \to 0$, any ℓ_p-norm $\|\mathbf{W}_1 - \mathbf{W}_2\|_p \to 0$ but $|\mathcal{NE}(\mathcal{G}_1)| - |\mathcal{NE}(\mathcal{G}_2)| = 2^n - 2$ remains constant.

33. The log-partition function of an Ising model is a trivial bound for counting the number of equilibria. To see this, let $f_i(\mathbf{x}_{-i}) \equiv \mathbf{w}_{i,-i}{}^{\mathrm{T}} \mathbf{x}_{-i} - b_i$, $|\mathcal{NE}(\mathcal{G})| = \sum_{\mathbf{x}} \prod_i 1[x_i f_i(\mathbf{x}_{-i}) \geq 0] \leq \sum_{\mathbf{x}} \prod_i e^{x_i f_i(\mathbf{x}_{-i})} = \sum_{\mathbf{x}} e^{\mathbf{x}^{\mathrm{T}} \mathbf{W} \mathbf{x} - \mathbf{b}^{\mathrm{T}} \mathbf{x}} = \mathcal{Z}(\frac{1}{2}(\mathbf{W} + \mathbf{W}^{\mathrm{T}}), \mathbf{b})$, where $\mathcal{Z}$ denotes the partition function of an Ising model. Given the convexity of $\mathcal{Z}$ (Koller and Friedman, 2009), and that the gradient vanishes at $\mathbf{W} = \mathbf{0}, \mathbf{b} = \mathbf{0}$, we know that $\mathcal{Z}(\frac{1}{2}(\mathbf{W} + \mathbf{W}^{\mathrm{T}}), \mathbf{b}) \geq 2^n$, which is the maximum $|\mathcal{NE}(\mathcal{G})|$.

unobserved joint action will increase the true proportion of equilibria without increasing the empirical proportion of equilibria and thus leading to a lower log-likelihood in Eq. (5). Second, given a fixed number of Nash equilibria k, the best strategy would be to pick the k joint actions that appear more frequently in the observed data. This will maximize the empirical proportion of equilibria, which will maximize the log-likelihood. Based on these observations, we propose Algorithm 1.

As an aside note, the fact that general games do not constrain the set of Nash equilibria, makes the method more likely to over-fit. On the other hand, LIGs will potentially include unobserved equilibria given the linearity constraints in the search space, and thus they would be less likely to over-fit.

6.2 An Exact Super-Exponential Method for LIGs: Exhaustive Search

Note that in the previous subsection, we search in the space of all possible games, not only the LIGs. First note that *sample-picking* for linear games is NP-hard, i.e., at any iteration of *sample-picking*, checking whether the set of Nash equilibria $\mathcal{NE}$ corresponds to an LIG or not is equivalent to the following constraint satisfaction problem with linear constraints:

$$\begin{aligned} &\min_{\mathbf{W},\mathbf{b}} \ 1 \\ &\text{s.t. } (\forall \mathbf{x} \in \mathcal{NE})\ x_1({\mathbf{w}_{1,-1}}^{\mathrm{T}}\mathbf{x}_{-1} - b_1) \geq 0 \wedge \cdots \wedge x_n({\mathbf{w}_{n,-n}}^{\mathrm{T}}\mathbf{x}_{-n} - b_n) \geq 0\ , \\ &\qquad (\forall \mathbf{x} \notin \mathcal{NE})\ x_1({\mathbf{w}_{1,-1}}^{\mathrm{T}}\mathbf{x}_{-1} - b_1) < 0 \vee \cdots \vee x_n({\mathbf{w}_{n,-n}}^{\mathrm{T}}\mathbf{x}_{-n} - b_n) < 0\ . \end{aligned} \tag{6}$$

Note that Eq. (6) contains "or" operators in order to account for the non-equilibria. This makes the problem of finding the $(\mathbf{W}, \mathbf{b})$ that satisfies such conditions NP-hard for a non-empty complement set $\{-1,+1\}^n - \mathcal{NE}$. Furthermore, since *sample-picking* only consider observed equilibria, the search is not optimal with respect to the space of LIGs.

Regarding a more refined approach for enumerating LIGs only, note that in an LIG each player separates hypercube vertices with a linear function, i.e., for $\mathbf{v} \equiv (\mathbf{w}_{i,-i}, b_i)$ and $\mathbf{y} \equiv (x_i\mathbf{x}_{-i}, -x_i) \in \{-1,+1\}^n$ we have $x_i({\mathbf{w}_{i,-i}}^{\mathrm{T}}\mathbf{x}_{-i} - b_i) = \mathbf{v}^{\mathrm{T}}\mathbf{y}$. Assume we assign a binary label to each vertex $\mathbf{y}$, then note that not all possible labelings are linearly separable. Labelings which are linearly separable are called *linear threshold functions (LTFs)*. A lower bound of the number of LTFs was first provided in Muroga (1965), which showed that the number of LTFs is at least $\alpha(n) \equiv 2^{0.33048n^2}$. Tighter lower bounds were shown later in Yamija and Ibaraki (1965) for $n \geq 6$ and in Muroga and Toda (1966) for $n \geq 8$. Regarding an upper bound, Winder (1960) showed that the number of LTFs is at most $\beta(n) \equiv 2^{n^2}$. By using such bounds for all players, we can conclude that there is at least $\alpha(n)^n = 2^{0.33048n^3}$ and at most $\beta(n)^n = 2^{n^3}$ LIGs (which is indeed another upper bound of the VC-dimension of the class of LIGs; the bound in Theorem 14 is tighter and uses bounds of the VC-dimension of neural networks). The bounds discussed above would bound the time-complexity of a search algorithm if we could easily enumerate all LTFs for a single player. Unfortunately, this seems to be far from a trivial problem. By using results in Muroga (1971), a weight vector $\mathbf{v}$ with integer entries such that $(\forall i)\ |v_i| \leq \beta(n) \equiv (n+1)^{(n+1)/2}/2^n$ is sufficient to realize all possible LTFs. Therefore we can conclude that enumerating LIGs takes at most $(2\beta(n)+1)^{n^2} \approx (\frac{\sqrt{n+1}}{2})^{n^3}$ steps, and we propose the use of this method only for $n \leq 4$.

For $n = 4$ we found that the number of possible PSNE sets induced by LIGs is 23,706. Experimentally, we did not find differences between this method and *sample-picking* since most of the time, the model with maximum likelihood was an LIG.

6.3 From Maximum Likelihood to Maximum Empirical Proportion of Equilibria

We approximately perform maximum likelihood estimation for LIGs, by maximizing the *empirical proportion of equilibria*, i.e., the equilibria in the observed data. This strategy allows us to avoid computing $\pi(\mathcal{G})$ as in Eq. (2) for maximum likelihood estimation (given its dependence on $|\mathcal{NE}(\mathcal{G})|$). We propose this approach for games with small true proportion of equilibria with high probability, i.e., with probability at least $1-\delta$, we have $\pi(\mathcal{G}) \leq \frac{\kappa^n}{\delta}$ for $1/2 \leq \kappa < 1$. Particularly, we will show in Section 7 that for LIGs we have $\kappa = 3/4$. Given this, our approximate problem relies on a bound of the log-likelihood that holds with high probability. We also show that under very mild conditions, the parameters $(\mathcal{G}, q)$ belong to the hypothesis space of the original problem with high probability.

First, we derive bounds on the log-likelihood function.

Lemma 16 *Given a non-trivial game $\mathcal{G}$ with $0 < \pi(\mathcal{G}) < \widehat{\pi}(\mathcal{G})$, the KL divergence in the log-likelihood function in Eq.* (5) *is bounded as follows:*

$$-\widehat{\pi}(\mathcal{G}) \log \pi(\mathcal{G}) - \log 2 < KL(\widehat{\pi}(\mathcal{G}) \| \pi(\mathcal{G})) < -\widehat{\pi}(\mathcal{G}) \log \pi(\mathcal{G}) \ .$$

Proof Let $\pi \equiv \pi(\mathcal{G})$ and $\widehat{\pi} \equiv \widehat{\pi}(\mathcal{G})$. Note that $\alpha(\pi) \equiv \lim_{\widehat{\pi} \to 0} KL(\widehat{\pi} \| \pi) = 0$,[34] and $\beta(\pi) \equiv \lim_{\widehat{\pi} \to 1} KL(\widehat{\pi} \| \pi) = -\log \pi \leq n \log 2$. Since the function is convex we can upper-bound it by $\alpha(\pi) + (\beta(\pi) - \alpha(\pi))\widehat{\pi} = -\widehat{\pi} \log \pi$.

To find a lower bound, we find the point in which the derivative of the original function is equal to the slope of the upper bound, i.e., $\frac{\partial KL(\widehat{\pi} \| \pi)}{\partial \widehat{\pi}} = \beta(\pi) - \alpha(\pi) = -\log \pi$, which gives $\widehat{\pi}^* = \frac{1}{2-\pi}$. Then, the maximum difference between the upper bound and the original function is given by $\lim_{\pi \to 0} -\widehat{\pi}^* \log \pi - KL(\widehat{\pi}^* \| \pi) = \log 2$. ■

Note that the lower and upper bounds are very informative when $\pi(\mathcal{G}) \to 0$ (or in our setting when $n \to +\infty$), since $\log 2$ becomes small when compared to $-\log \pi(\mathcal{G})$, as shown in Figure 2.

Next, we derive the problem of maximizing the empirical proportion of equilibria from the maximum likelihood estimation problem.

Theorem 17 *Assume that with probability at least $1-\delta$ we have $\pi(\mathcal{G}) \leq \frac{\kappa^n}{\delta}$ for $1/2 \leq \kappa < 1$. Maximizing a lower bound (with high probability) of the log-likelihood in Eq.* (5) *is equivalent*

34. Here we are making the implicit assumption that $\pi < \widehat{\pi}$. This is sensible. For example, in most models learned from the congressional voting data using a variety of learning algorithms we propose, the total number of PSNE would range roughly from 100K—1M; using base 2, this is roughly from 2^{16}—2^{20}. This may look like a huge number until one recognizes that there could potential be 2^{100} PSNE. Hence, we have that π would be in the range of 2^{-84}—2^{-80}. In fact, we believe this holds more broadly because, as a general objective, we want models that can capture as many PSNE behavior as possible but no more than needed, which tend to reduce the PSNE of the learned models, and thus their π values, while simultaneously trying to increase $\widehat{\pi}$ as much as possible.

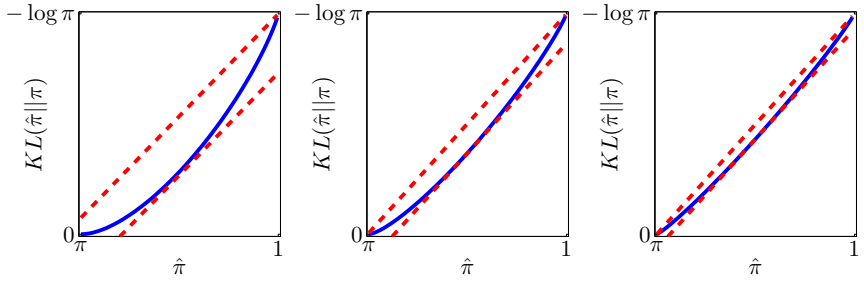


Figure 2: KL divergence (blue) and bounds derived in Lemma 16 (red) for $\pi = (3/4)^n$ where $n = 9$ (left), $n = 18$ (center) and $n = 36$ (right). Note that the bounds are very informative when $n \to +\infty$ (or equivalently when $\pi \to 0$).

to maximizing the empirical proportion of equilibria:

$$\max_{\mathcal{G}\in\mathcal{H}} \widehat{\pi}(\mathcal{G}) \,, \tag{7}$$

furthermore, for all games $\mathcal{G}$ such that $\widehat{\pi}(\mathcal{G}) \geq \gamma$ for some $0 < \gamma < 1/2$, for sufficiently large $n > \log_\kappa(\delta\gamma)$ and optimal mixture parameter $\widehat{q} = \min(\widehat{\pi}(\mathcal{G}), 1 - \frac{1}{2m})$, we have $(\mathcal{G}, \widehat{q}) \in \Upsilon$, where $\Upsilon = \{(\mathcal{G}, q) \mid \mathcal{G} \in \mathcal{H} \wedge 0 < \pi(\mathcal{G}) < q < 1\}$ is the hypothesis space of non-trivial identifiable games and mixture parameters.

Proof By applying the lower bound in Lemma 16 in Eq. (5) to non-trivial games, we have $\widehat{\mathcal{L}}(\mathcal{G}, \widehat{q}) = KL(\widehat{\pi}(\mathcal{G})\|\pi(\mathcal{G})) - KL(\widehat{\pi}(\mathcal{G})\|\widehat{q}) - n\log 2 > -\widehat{\pi}(\mathcal{G})\log\pi(\mathcal{G}) - KL(\widehat{\pi}(\mathcal{G})\|\widehat{q}) - (n+1)\log 2$. Since $\pi(\mathcal{G}) \leq \frac{\kappa^n}{\delta}$, we have $-\log\pi(\mathcal{G}) \geq -\log\frac{\kappa^n}{\delta}$. Therefore $\widehat{\mathcal{L}}(\mathcal{G}, \widehat{q}) > -\widehat{\pi}(\mathcal{G})\log\frac{\kappa^n}{\delta} - KL(\widehat{\pi}(\mathcal{G})\|\widehat{q}) - (n+1)\log 2$. Regarding the term $KL(\widehat{\pi}(\mathcal{G})\|\widehat{q})$, if $\widehat{\pi}(\mathcal{G}) < 1 \Rightarrow KL(\widehat{\pi}(\mathcal{G})\|\widehat{q}) = KL(\widehat{\pi}(\mathcal{G})\|\widehat{\pi}(\mathcal{G})) = 0$, and if $\widehat{\pi}(\mathcal{G}) = 1 \Rightarrow KL(\widehat{\pi}(\mathcal{G})\|\widehat{q}) = KL(1\|1 - \frac{1}{2m}) = -\log(1 - \frac{1}{2m}) \leq \log 2$ and approaches 0 when $m \to +\infty$. Maximizing the lower bound of the log-likelihood becomes $\max_{\mathcal{G}\in\mathcal{H}} \widehat{\pi}(\mathcal{G})$ by removing the constant terms that do not depend on $\mathcal{G}$.

In order to prove $(\mathcal{G}, \widehat{q}) \in \Upsilon$ we need to prove $0 < \pi(\mathcal{G}) < \widehat{q} < 1$. For proving the first inequality $0 < \pi(\mathcal{G})$, note that $\widehat{\pi}(\mathcal{G}) \geq \gamma > 0$, and therefore $\mathcal{G}$ has at least one equilibria. For proving the third inequality $\widehat{q} < 1$, note that $\widehat{q} = \min(\widehat{\pi}(\mathcal{G}), 1 - \frac{1}{2m}) < 1$. For proving the second inequality $\pi(\mathcal{G}) < \widehat{q}$, we need to prove $\pi(\mathcal{G}) < \widehat{\pi}(\mathcal{G})$ and $\pi(\mathcal{G}) < 1 - \frac{1}{2m}$. Since $\pi(\mathcal{G}) \leq \frac{\kappa^n}{\delta}$ and $\gamma \leq \widehat{\pi}(\mathcal{G})$, it suffices to prove $\frac{\kappa^n}{\delta} < \gamma \Rightarrow \pi(\mathcal{G}) < \widehat{\pi}(\mathcal{G})$. Similarly we need to prove $\frac{\kappa^n}{\delta} < 1 - \frac{1}{2m} \Rightarrow \pi(\mathcal{G}) < 1 - \frac{1}{2m}$. Putting both together, we have $\frac{\kappa^n}{\delta} < \min(\gamma, 1 - \frac{1}{2m}) = \gamma$ since $\gamma < 1/2$ and $1 - \frac{1}{2m} \geq 1/2$. Finally, $\frac{\kappa^n}{\delta} < \gamma \Leftrightarrow n > \log_\kappa(\delta\gamma)$. ■

6.4 A Non-Concave Maximization Method: Sigmoidal Approximation

A very simple optimization approach can be devised by using a sigmoid in order to approximate the 0/1 function $1[z \geq 0]$ in the maximum likelihood problem of Eq. (5) as well as when maximizing the empirical proportion of equilibria as in Eq. (7). We use the following

sigmoidal approximation:

$$1[z \geq 0] \approx H_{\alpha,\beta}(z) \equiv \tfrac{1}{2}(1 + \tanh(\tfrac{z}{\beta} - \operatorname{arctanh}(1 - 2\alpha^{1/n}))) \ . \tag{8}$$

The additional term α ensures that for $\mathcal{G} = (\mathbf{W}, \mathbf{b}), \mathbf{W} = \mathbf{0}, \mathbf{b} = \mathbf{0}$ we get $1[\mathbf{x} \in \mathcal{NE}(\mathcal{G})] \approx H_{\alpha,\beta}(0)^n = \alpha$. We perform gradient ascent on these objective functions that have many local maxima. Note that when maximizing the "sigmoidal" likelihood, each step of the gradient ascent is NP-hard due to the "sigmoidal" true proportion of equilibria. Therefore, we propose the use of the sigmoidal maximum likelihood only for $n \leq 15$.

In our implementation, we add an ℓ_1-norm regularizer $-\rho\|\mathbf{W}\|_1$ where $\rho > 0$ to both maximization problems. The ℓ_1-norm regularizer encourages sparseness and attempts to lower the generalization error by controlling over-fitting.

6.5 Our Proposed Approach: Convex Loss Minimization

From an optimization perspective, it is more convenient to minimize a convex objective instead of a sigmoidal approximation in order to avoid the many local minima.

Note that maximizing the empirical proportion of equilibria in Eq. (7) is equivalent to minimizing the empirical proportion of non-equilibria, i.e., $\min_{\mathcal{G} \in \mathcal{H}} (1 - \widehat{\pi}(\mathcal{G}))$. Furthermore, $1 - \widehat{\pi}(\mathcal{G}) = \frac{1}{m} \sum_l 1[\mathbf{x}^{(l)} \notin \mathcal{NE}(\mathcal{G})]$. Denote by ℓ the 0/1 loss, i.e., $\ell(z) = 1[z < 0]$. For LIGs, maximizing the empirical proportion of equilibria in Eq. (7) is equivalent to solving the loss minimization problem:

$$\min_{\mathbf{W},\mathbf{b}} \ \frac{1}{m} \sum_l \max_i \ell(x_i^{(l)}(\mathbf{w}_{i,-i}{}^{\mathrm{T}} \mathbf{x}_{-i}^{(l)} - b_i)) \ . \tag{9}$$

We can further relax this problem by introducing convex upper bounds of the 0/1 loss. Note that the use of convex losses also avoids the trivial solution of Eq. (9), i.e., $\mathbf{W} = \mathbf{0}, \mathbf{b} = \mathbf{0}$ (which obtains the lowest log-likelihood as discussed in Remark 10). Intuitively speaking, note that minimizing the logistic loss $\ell(z) = \log(1 + e^{-z})$ will make $z \to +\infty$, while minimizing the hinge loss $\ell(z) = \max(0, 1 - z)$ will make $z \to 1$ unlike the 0/1 loss $\ell(z) = 1[z < 0]$ that only requires $z = 0$ in order to be minimized. In what follows, we develop four efficient methods for solving Eq. (9) under specific choices of loss functions, i.e., hinge and logistic.

In our implementation, we add an ℓ_1-norm regularizer $\rho\|\mathbf{W}\|_1$ where $\rho > 0$ to all the minimization problems. The ℓ_1-norm regularizer encourages sparseness and attempts to lower the generalization error by controlling over-fitting.

6.5.1 Independent Support Vector Machines and Logistic Regression

We can relax the loss minimization problem in Eq. (9) by using the loose bound $\max_i \ell(z_i) \leq \sum_i \ell(z_i)$. This relaxation simplifies the original problem into several independent problems. For each player i, we train the weights $(\mathbf{w}_{i,-i}, b_i)$ in order to predict independent (disjoint) actions. This leads to *1-norm SVMs* of Bradley and Mangasarian (1998); Zhu et al. (2004) and ℓ_1-regularized logistic regression. We solve the latter with the *ℓ_1-projection method* of Schmidt et al. (2007a). While the training is independent, our goal is not the prediction for independent players but the characterization of joint actions. The use of these

well known techniques in our context is novel, since we interpret the output of SVMs and logistic regression as the parameters of an LIG. Therefore, we use the parameters to measure empirical and true proportion of equilibria, KL divergence and log-likelihood in our probabilistic model.

6.5.2 Simultaneous Support Vector Machines

While converting the loss minimization problem in Eq. (9) by using loose bounds allow to obtain several independent problems with small number of variables, a second reasonable strategy would be to use tighter bounds at the expense of obtaining a single optimization problem with a higher number of variables.

For the hinge loss $\ell(z) = \max(0, 1 - z)$, we have $\max_i \ell(z_i) = \max(0, 1 - z_1, \ldots, 1 - z_n)$ and the loss minimization problem in Eq. (9) becomes the following primal linear program:

$$\begin{aligned} &\min_{\mathbf{W},\mathbf{b},\boldsymbol{\xi}} \ \frac{1}{m}\sum_l \xi_l + \rho\|\mathbf{W}\|_1 \\ &\text{s.t. } (\forall l, i)\ x_i^{(l)}({\mathbf{w}_{i,-i}}^{\mathrm{T}}\mathbf{x}_{-i}^{(l)} - b_i) \geq 1 - \xi_l \ \ , \ \ (\forall l)\ \xi_l \geq 0\ , \end{aligned} \tag{10}$$

where $\rho > 0$.

Note that Eq. (10) is equivalent to a linear program since we can set $\mathbf{W} = \mathbf{W}^+ - \mathbf{W}^-$, $\|\mathbf{W}\|_1 = \sum_{ij} w_{ij}^+ + w_{ij}^-$ and add the constraints $\mathbf{W}^+ \geq \mathbf{0}$ and $\mathbf{W}^- \geq \mathbf{0}$. We follow the regular SVM derivation by adding slack variables ξ_l for each sample l. This problem is a generalization of *1-norm SVMs* of Bradley and Mangasarian (1998); Zhu et al. (2004).

By Lagrangian duality, the dual of the problem in Eq. (10) is the following linear program:

$$\begin{aligned} &\max_{\boldsymbol{\alpha}} \ \sum_{li} \alpha_{li} \\ &\text{s.t. } (\forall i)\ \|\textstyle\sum_l \alpha_{li} x_i^{(l)} \mathbf{x}_{-i}^{(l)}\|_\infty \leq \rho \ \ , \ \ (\forall l, i)\ \alpha_{li} \geq 0\ , \\ &\qquad (\forall i)\ \textstyle\sum_l \alpha_{li} x_i^{(l)} = 0\ , \qquad\qquad (\forall l)\ \sum_i \alpha_{li} \leq \frac{1}{m}\ . \end{aligned} \tag{11}$$

Furthermore, strong duality holds in this case. Note that Eq. (11) is equivalent to a linear program since we can transform the constraint $\|\mathbf{c}\|_\infty \leq \rho$ into $-\rho\mathbf{1} \leq \mathbf{c} \leq \rho\mathbf{1}$.

6.5.3 Simultaneous Logistic Regression

For the logistic loss $\ell(z) = \log(1+e^{-z})$, we could use the non-smooth loss $\max_i \ell(z_i)$ directly. Instead, we chose a smooth upper bound, i.e., $\log(1 + \sum_i e^{-z_i})$. The following discussion and technical lemma provides the reason behind our us of this *simultaneous logistic loss*.

Given that any loss $\ell(z)$ is a decreasing function, the following identity holds $\max_i \ell(z_i) = \ell(\min_i z_i)$. Hence, we can either upper-bound the max function by the logsumexp function or lower-bound the min function by a negative logsumexp. We chose the latter option for the logistic loss for the following reasons: Claim i of the following technical lemma shows that lower-bounding min generates a loss that is strictly less than upper-bounding max. Claim ii shows that lower-bounding min generates a loss that is strictly less than independently penalizing each player. Claim iii shows that there are some cases in which upper-bounding max generates a loss that is strictly greater than independently penalizing each player.

Lemma 18 *For the logistic loss* $\ell(z) = \log(1+e^{-z})$ *and a set of* $n > 1$ *numbers* $\{z_1, \ldots, z_n\}$*:*

i. $(\forall z_1, \ldots, z_n)\ \max_i \ell(z_i) \leq \ell\left(-\log \sum_i e^{-z_i}\right) < \log \sum_i e^{\ell(z_i)} \leq \max_i \ell(z_i) + \log n$,
ii. $(\forall z_1, \ldots, z_n)\ \ell\left(-\log \sum_i e^{-z_i}\right) < \sum_i \ell(z_i)$,
iii. $(\exists z_1, \ldots, z_n)\ \log \sum_i e^{\ell(z_i)} > \sum_i \ell(z_i)$.

Proof Given a set of numbers $\{a_1, \ldots, a_n\}$, the max function is bounded by the logsumexp function by $\max_i a_i \leq \log \sum_i e^{a_i} \leq \max_i a_i + \log n$ (Boyd and Vandenberghe, 2006). Equivalently, the min function is bounded by $\min_i a_i - \log n \leq -\log \sum_i e^{-a_i} \leq \min_i a_i$.

These identities allow us to prove two inequalities in Claim i, i.e., $\max_i \ell(z_i) = \ell(\min_i z_i) \leq \ell\left(-\log \sum_i e^{-z_i}\right)$ and $\log \sum_i e^{\ell(z_i)} \leq \max_i \ell(z_i) + \log n$. To prove the remaining inequality $\ell\left(-\log \sum_i e^{-z_i}\right) < \log \sum_i e^{\ell(z_i)}$, note that for the logistic loss $\ell\left(-\log \sum_i e^{-z_i}\right) = \log(1 + \sum_i e^{-z_i})$ and $\log \sum_i e^{\ell(z_i)} = \log(n + \sum_i e^{-z_i})$. Since $n > 1$, strict inequality holds.

To prove Claim ii, we need to show that $\ell\left(-\log \sum_i e^{-z_i}\right) = \log(1+\sum_i e^{-z_i}) < \sum_i \ell(z_i) = \sum_i \log(1 + e^{-z_i})$. This is equivalent to $1 + \sum_i e^{-z_i} < \prod_i (1 + e^{-z_i}) = \sum_{\mathbf{c} \in \{0,1\}^n} e^{-\mathbf{c}^{\mathrm{T}}\mathbf{z}} = 1 + \sum_i e^{-z_i} + \sum_{\mathbf{c} \in \{0,1\}^n, \mathbf{1}^{\mathrm{T}}\mathbf{c} > 1} e^{-\mathbf{c}^{\mathrm{T}}\mathbf{z}}$. Finally, we have $\sum_{\mathbf{c} \in \{0,1\}^n, \mathbf{1}^{\mathrm{T}}\mathbf{c} > 1} e^{-\mathbf{c}^{\mathrm{T}}\mathbf{z}} > 0$ because the exponential function is strictly positive.

To prove Claim iii, it suffices to find set of numbers $\{z_1, \ldots, z_n\}$ for which $\log \sum_i e^{\ell(z_i)} = \log(n + \sum_i e^{-z_i}) > \sum_i \ell(z_i) = \sum_i \log(1 + e^{-z_i})$. This is equivalent to $n + \sum_i e^{-z_i} > \prod_i (1 + e^{-z_i})$. By setting $(\forall i)\ z_i = \log n$, we reduce the claim we want to prove to $n + 1 > (1+\frac{1}{n})^n$. Strict inequality holds for $n > 1$. Furthermore, note that $\lim_{n \to +\infty} (1 + \frac{1}{n})^n = e$. ■

Returning to our simultaneous logistic regression formulation, the loss minimization problem in Eq. (9) becomes

$$\min_{\mathbf{W}, \mathbf{b}} \ \frac{1}{m} \sum_l \log(1 + \textstyle\sum_i e^{-x_i^{(l)}(\mathbf{w}_{i,-i}{}^{\mathrm{T}}\mathbf{x}_{-i}^{(l)} - b_i)}) + \rho \|\mathbf{W}\|_1 \ , \tag{12}$$

where $\rho > 0$.

In our implementation, we use the ℓ_1-*projection method* of Schmidt et al. (2007a) for optimizing Eq. (12). This method performs a *limited-memory Broyden-Fletcher-Goldfarb-Shanno* (L-BFGS) step in an expanded model (i.e., $\mathbf{W} = \mathbf{W}^+ - \mathbf{W}^-$, $\|\mathbf{W}\|_1 = \sum_{ij} w_{ij}^+ + w_{ij}^-$) followed by a projection onto the non-negative orthant to enforce $\mathbf{W}^+ \geq \mathbf{0}$ and $\mathbf{W}^- \geq \mathbf{0}$.

7. On the True Proportion of Equilibria

In this section, we justify the use of convex loss minimization for learning the structure and parameters of LIGs. We define *absolute indifference* of players and show that our convex loss minimization approach produces games in which all players are non-absolutely-indifferent. We then provide a bound of the true proportion of equilibria with high probability. Our bound assumes independence of weight vectors among players, and applies to a large family of distributions of weight vectors. Furthermore, we do not assume any connectivity properties of the underlying graph.

Parallel to our analysis, Daskalakis et al. (2011) analyzed a different setting: random games which structure is drawn from the Erdős-Rényi model (i.e., each edge is present

independently with the same probability p) and utility functions which are random tables. The analysis in Daskalakis et al. (2011), while more general than ours (which only focus on LIGs), it is at the same time more restricted since it assumes either the Erdős-Rényi model for random structures or connectivity properties for deterministic structures.

7.1 Convex Loss Minimization Produces Non-Absolutely-Indifferent Players

First, we define the notion of *absolute indifference* of players. Our goal in this subsection is to show that our proposed convex loss algorithms produce LIGs in which all players are non-absolutely-indifferent and therefore every player defines constraints to the true proportion of equilibria.

Definition 19 *Given an LIG $\mathcal{G} = (\mathbf{W}, \mathbf{b})$, we say a player i is* absolutely indifferent *if and only if $(\mathbf{w}_{i,-i}, b_i) = \mathbf{0}$, and* non-absolutely-indifferent *if and only if $(\mathbf{w}_{i,-i}, b_i) \neq \mathbf{0}$.*

Next, we concentrate on the first ingredient for our bound of the true proportion of equilibria. We show that independent and simultaneous SVM and logistic regression produce games in which all players are non-absolutely-indifferent except for some "degenerate" cases. The following lemma applies to independent SVMs for $c^{(l)} = 0$ and simultaneous SVMs for $c^{(l)} = \max(0, \max_{j \neq i} (1 - x_j^{(l)}(\mathbf{w}_{i,-i}{}^{\mathrm{T}} \mathbf{x}_{-i}^{(l)} - b_i)))$.

Lemma 20 *Given $(\forall l)\ c^{(l)} \geq 0$, the minimization of the hinge training loss $\widehat{\ell}(\mathbf{w}_{i,-i}, b_i) = \frac{1}{m} \sum_l \max(c^{(l)}, 1 - x_i^{(l)}(\mathbf{w}_{i,-i}{}^{\mathrm{T}} \mathbf{x}_{-i}^{(l)} - b_i))$ guarantees non-absolutely-indifference of player i except for some "degenerate" cases, i.e., the optimal solution $(\mathbf{w}_{i,-i}^*, b_i^*) = \mathbf{0}$ if and only if $(\forall j \neq i)\ \sum_l 1[x_i^{(l)} x_j^{(l)} = 1] u^{(l)} = \sum_l 1[x_i^{(l)} x_j^{(l)} = -1] u^{(l)}$ and $\sum_l 1[x_i^{(l)} = 1] u^{(l)} = \sum_l 1[x_i^{(l)} = -1] u^{(l)}$ where $u^{(l)}$ is defined as $c^{(l)} > 1 \Leftrightarrow u^{(l)} = 0$, $c^{(l)} < 1 \Leftrightarrow u^{(l)} = 1$ and $c^{(l)} = 1 \Leftrightarrow u^{(l)} \in [0; 1]$.*

Proof Let $f_i(\mathbf{x}_{-i}) \equiv \mathbf{w}_{i,-i}{}^{\mathrm{T}} \mathbf{x}_{-i} - b_i$. By noting that $\max(\alpha, \beta) = \max_{0 \leq u \leq 1} (\alpha + u(\beta - \alpha))$, we can rewrite $\widehat{\ell}(\mathbf{w}_{i,-i}, b_i) = \frac{1}{m} \sum_l \max_{0 \leq u^{(l)} \leq 1} (c^{(l)} + u^{(l)}(1 - x_i^{(l)} f_i(\mathbf{x}_{-i}^{(l)}) - c^{(l)}))$.

Note that $\widehat{\ell}$ has the minimizer $(\mathbf{w}_{i,-i}^*, b_i^*) = \mathbf{0}$ if and only if $\mathbf{0}$ belongs to the subdifferential set of the non-smooth function $\widehat{\ell}$ at $(\mathbf{w}_{i,-i}, b_i) = \mathbf{0}$. In order to maximize $\widehat{\ell}$, we have $c^{(l)} > 1 - x_i^{(l)} f_i(\mathbf{x}_{-i}^{(l)}) \Leftrightarrow u^{(l)} = 0$, $c^{(l)} < 1 - x_i^{(l)} f_i(\mathbf{x}_{-i}^{(l)}) \Leftrightarrow u^{(l)} = 1$ and $c^{(l)} = 1 - x_i^{(l)} f_i(\mathbf{x}_{-i}^{(l)}) \Leftrightarrow u^{(l)} \in [0; 1]$. The previous rules simplify at the solution under analysis, since $(\mathbf{w}_{i,-i}, b_i) = \mathbf{0} \Rightarrow f_i(\mathbf{x}_{-i}^{(l)}) = 0$.

Let $g_j(\mathbf{w}_{i,-i}, b_i) \equiv \frac{\partial \widehat{\ell}}{\partial w_{ij}}(\mathbf{w}_{i,-i}, b_i)$ and $h(\mathbf{w}_{i,-i}, b_i) \equiv \frac{\partial \widehat{\ell}}{\partial b_i}(\mathbf{w}_{i,-i}, b_i)$. By making $(\forall j \neq i)\ 0 \in g_j(\mathbf{0}, 0)$ and $0 \in h(\mathbf{0}, 0)$, we get $(\forall j \neq i)\ \sum_l x_i^{(l)} x_j^{(l)} u^{(l)} = 0$ and $\sum_l x_i^{(l)} u^{(l)} = 0$. Finally, by noting that $x_i^{(l)} \in \{-1, 1\}$, we prove our claim. ■

Remark 21 *Note that for independent SVMs, the "degenerate" cases in Lemma 20 simplify to $(\forall j \neq i)\ \sum_l 1[x_i^{(l)} x_j^{(l)} = 1] = \frac{m}{2}$ and $\sum_l 1[x_i^{(l)} = 1] = \frac{m}{2}$.*

The following lemma applies to independent logistic regression for $c^{(l)} = 0$ and simultaneous logistic regression for $c^{(l)} = \sum_{j \neq i} e^{-x_j^{(l)}(\mathbf{w}_{i,-i}{}^{\mathrm{T}} \mathbf{x}_{-i}^{(l)} - b_i)}$.

Lemma 22 *Given* $(\forall l)\ c^{(l)} \geq 0$*, the minimization of the logistic training loss* $\widehat{\ell}(\mathbf{w}_{i,-i}, b_i) = \frac{1}{m}\sum_l \log(c^{(l)} + 1 + e^{-x_i^{(l)}({\mathbf{w}_{i,-i}}^{\mathrm{T}}\mathbf{x}_{-i}^{(l)} - b_i)})$ *guarantees non-absolutely-indifference of player* i *except for some "degenerate" cases, i.e., the optimal solution* $(\mathbf{w}_{i,-i}^*, b_i^*) = \mathbf{0}$ *if and only if* $(\forall j \neq i)\ \sum_l \frac{1[x_i^{(l)} x_j^{(l)} = 1]}{c^{(l)}+2} = \sum_l \frac{1[x_i^{(l)} x_j^{(l)} = -1]}{c^{(l)}+2}$ *and* $\sum_l \frac{1[x_i^{(l)}=1]}{c^{(l)}+2} = \sum_l \frac{1[x_i^{(l)}=-1]}{c^{(l)}+2}$.

Proof Note that $\widehat{\ell}$ has the minimizer $(\mathbf{w}_{i,-i}^*, b_i^*) = \mathbf{0}$ if and only if the gradient of the smooth function $\widehat{\ell}$ is $\mathbf{0}$ at $(\mathbf{w}_{i,-i}, b_i) = \mathbf{0}$. Let $g_j(\mathbf{w}_{i,-i}, b_i) \equiv \frac{\partial \widehat{\ell}}{\partial w_{ij}}(\mathbf{w}_{i,-i}, b_i)$ and $h(\mathbf{w}_{i,-i}, b_i) \equiv \frac{\partial \widehat{\ell}}{\partial b_i}(\mathbf{w}_{i,-i}, b_i)$. By making $(\forall j \neq i)\ g_j(\mathbf{0}, 0) = 0$ and $h(\mathbf{0}, 0) = 0$, we get $(\forall j \neq i)\ \sum_l \frac{x_i^{(l)} x_j^{(l)}}{c^{(l)}+2} = 0$ and $\sum_l \frac{x_i^{(l)}}{c^{(l)}+2} = 0$. Finally, by noting that $x_i^{(l)} \in \{-1, 1\}$, we prove our claim. ■

Remark 23 *Note that for independent logistic regression, the "degenerate" cases in Lemma 22 simplify to* $(\forall j \neq i)\ \sum_l 1[x_i^{(l)} x_j^{(l)} = 1] = \frac{m}{2}$ *and* $\sum_l 1[x_i^{(l)} = 1] = \frac{m}{2}$.

Based on these results, after termination of our proposed algorithms, we fix cases in which the optimal solution $(\mathbf{w}_{i,-i}^*, b_i^*) = \mathbf{0}$ by setting $b_i^* = 1$ if the action of player i was mostly -1 or $b_i^* = -1$ otherwise. We point out to the careful reader that we did not include the ℓ_1-regularization term in the above proofs since the subdifferential of $\rho\|\mathbf{w}_{i,-i}\|_1$ vanishes at $\mathbf{w}_{i,-i} = 0$, and therefore our proofs still hold.

7.2 Bounding the True Proportion of Equilibria

In what follows, we concentrate on the second ingredient for our bound of the true proportion of equilibria. We show a bound for a single *non-absolutely-indifferent* player and a fixed joint-action $\mathbf{x}$, that interestingly does not depend on the specific joint-action $\mathbf{x}$. This is a key ingredient for bounding the true proportion of equilibria in our main theorem.

Lemma 24 *Given an LIG* $\mathcal{G} = (\mathbf{W}, \mathbf{b})$ *with non-absolutely-indifferent player* i*, assume that* $(\mathbf{w}_{i,-i}, b_i)$ *is a random vector drawn from a distribution* $\mathcal{P}_i$*. If for all* $\mathbf{x} \in \{-1, +1\}^n$, $\mathbb{P}_{\mathcal{P}_i}[x_i({\mathbf{w}_{i,-i}}^{\mathrm{T}}\mathbf{x}_{-i} - b_i) = 0] = 0$*, then*

$$\textit{i. for all } \mathbf{x},\ \mathbb{P}_{\mathcal{P}_i}[x_i({\mathbf{w}_{i,-i}}^{\mathrm{T}}\mathbf{x}_{-i} - b_i) \geq 0] = \mathbb{P}_{\mathcal{P}_i}[x_i({\mathbf{w}_{i,-i}}^{\mathrm{T}}\mathbf{x}_{-i} - b_i) \leq 0]$$
$$\textit{if and only if, for all } \mathbf{x},\ \mathbb{P}_{\mathcal{P}_i}[x_i({\mathbf{w}_{i,-i}}^{\mathrm{T}}\mathbf{x}_{-i} - b_i) \geq 0] = 1/2\ .$$

If $\mathcal{P}_i$ *is a uniform distribution of support* $\{-1, +1\}^n$*, then*

$$\textit{ii. for all } \mathbf{x},\ \mathbb{P}_{\mathcal{P}_i}[x_i({\mathbf{w}_{i,-i}}^{\mathrm{T}}\mathbf{x}_{-i} - b_i) \geq 0] \in [1/2,\ 3/4]\ .$$

Proof Claim i follows immediately from a simple condition we obtain from the normalization axiom of probability and the hypothesis of the claim: i.e., for all $\mathbf{x} \in \{-1, +1\}^n$, $\mathbb{P}_{\mathcal{P}_i}[x_i({\mathbf{w}_{i,-i}}^{\mathrm{T}}\mathbf{x}_{-i} - b_i) \geq 0] + \mathbb{P}_{\mathcal{P}_i}[x_i({\mathbf{w}_{i,-i}}^{\mathrm{T}}\mathbf{x}_{-i} - b_i) \leq 0] = 1$.

To prove Claim ii, first let $\mathbf{v} \equiv (\mathbf{w}_{i,-i}, b_i)$ and $\mathbf{y} \equiv (x_i\mathbf{x}_{-i}, -x_i) \in \{-1, +1\}^n$. Note that $x_i({\mathbf{w}_{i,-i}}^{\mathrm{T}}\mathbf{x}_{-i} - b_i) = \mathbf{v}^{\mathrm{T}}\mathbf{y}$. Then, let $f_1(\mathbf{v}_{-1}, \mathbf{y}) \equiv {\mathbf{v}_{-1}}^{\mathrm{T}}\mathbf{y}_{-1} + y_1$. Note that $(v_1, v_1\mathbf{v}_{-1})$

spans all possible vectors in $\{-1,+1\}^n$. Because $\mathcal{P}_i$ is a uniform distribution of support $\{-1,+1\}^n$, we have:

$$\begin{aligned}
\mathbb{P}_{\mathcal{P}_i}[\mathbf{v}^{\mathrm{T}}\mathbf{y} \geq 0] &= \tfrac{1}{2^n}\textstyle\sum_{\mathbf{v}} 1[\mathbf{v}^{\mathrm{T}}\mathbf{y} \geq 0] \\
&= \tfrac{1}{2^n}\textstyle\sum_{\mathbf{v}} 1[v_1 f_1(\mathbf{v}_{-1},\mathbf{y}) \geq 0] \\
&= \tfrac{1}{2^n}\textstyle\sum_{\mathbf{v}} \left(1[v_1=+1]1[f_1(\mathbf{v}_{-1},\mathbf{y}) \geq 0] + 1[v_1=-1]1[f_1(\mathbf{v}_{-1},\mathbf{y}) \leq 0]\right) \\
&= \tfrac{1}{2^n}\textstyle\sum_{\mathbf{v}_{-1}} \left(1[f_1(\mathbf{v}_{-1},\mathbf{y}) \geq 0] + 1[f_1(\mathbf{v}_{-1},\mathbf{y}) \leq 0]\right) \\
&= \tfrac{2^{n-1}}{2^n} + \tfrac{1}{2^n}\textstyle\sum_{\mathbf{v}_{-1}} 1[f_1(\mathbf{v}_{-1},\mathbf{y}) = 0] \\
&= 1/2 + \tfrac{1}{2^n}\alpha(\mathbf{y})
\end{aligned}$$

where $\alpha(\mathbf{y}) \equiv \sum_{\mathbf{v}_{-1}} 1[f_1(\mathbf{v}_{-1},\mathbf{y}) = 0] = \sum_{\mathbf{v}_{-1}} 1[{\mathbf{v}_{-1}}^{\mathrm{T}}\mathbf{y}_{-1} + y_1 = 0]$. Note that $\alpha(\mathbf{y}) \geq 0$ and thus, $\mathbb{P}_{\mathcal{P}_i}[\mathbf{v}^{\mathrm{T}}\mathbf{y} \geq 0] \geq 1/2$. Geometrically speaking, $\alpha(\mathbf{y})$ is the number of vertices of the $(n-1)$-dimensional hypercube that are covered by the hyperplane with normal $\mathbf{y}_{-1}$ and bias y_1. Recall that $\mathbf{y} \neq \mathbf{0}$ since $\mathbf{y} \in \{-1,+1\}^n$. By relaxing this fact, as noted in Aichholzer and Aurenhammer (1996) a hyperplane with $n-2$ zeros on $\mathbf{y}_{-1}$ (i.e., a $(n-2)$-*parallel hyperplane*) covers exactly half of the 2^{n-1} vertices, the maximum possible. Therefore, $\mathbb{P}_{\mathcal{P}_i}[\mathbf{v}^{\mathrm{T}}\mathbf{y} \geq 0] = 1/2 + \frac{1}{2^n}\alpha(\mathbf{y}) \leq 1/2 + \frac{2^{n-2}}{2^n} = 3/4$. ■

Remark 25 *It is important to note that under the conditions of Lemma 24, in a measure-theoretic sense, for almost all vectors $(\mathbf{w}_{i,-i}, b_i)$ in the surface of the hypersphere in n-dimensions (i.e., except for a set of Lebesgue-measure zero), we have that, $x_i({\mathbf{w}_{i,-i}}^{\mathrm{T}}\mathbf{x}_{-i} - b_i) \neq 0$ for all $\mathbf{x} \in \{-1,+1\}^n$. Hence, the hypothesis stated for Claim i of Lemma 24 holds for almost all probability measures $\mathcal{P}_i$ (i.e., except for a set of probability measures, over the surface of the hypersphere in n-dimensions, with Lebesgue-measure zero). Note that Claim ii essentially states that we can still upper bound, for all $\mathbf{x} \in \{-1,+1\}^n$, the probability that such $\mathbf{x}$ is a PSNE of a random LIG even if we draw the weights and threshold parameters from a $\mathcal{P}_i$ belonging to such sets of Lebesgue-measure zero.*

Remark 26 *Note that any distribution that has zero mean and that depends on some norm of $(\mathbf{w}_{i,-i}, b_i)$ fulfills the requirements for Claim i of Lemma 24. This includes, for instance, the multivariate normal distribution with arbitrary covariance which is related to the Bhattacharyya norm. Additionally, any distribution in which each entry of the vector $(\mathbf{w}_{i,-i}, b_i)$ is independent and symmetric also fulfills those requirements. This includes, for instance, the Laplace and uniform distributions. Furthermore, note that distributions with support on non-empty subsets of entries of $(\mathbf{w}_{i,-i}, b_i)$, as well as mixtures of the above cases are also allowed. This includes, for instance, sparse graphs.*

Next, we present our bound for the true proportion of equilibria of games in which all players are non-absolutely-indifferent.

Theorem 27 *Assume that all players are non-absolutely-indifferent and that the rows of an LIG $\mathcal{G} = (\mathbf{W}, \mathbf{b})$ are independent (but not necessarily identically distributed) random*

vectors, i.e., for every player i, $(\mathbf{w}_{i,-i}, b_i)$ is independently drawn from an arbitrary distribution $\mathcal{P}_i$. If for all i and $\mathbf{x}$, $\mathbb{P}_{\mathcal{P}_i}[x_i({\mathbf{w}_{i,-i}}^{\mathrm{T}}\mathbf{x}_{-i} - b_i) \geq 0] \leq \kappa$ for $1/2 \leq \kappa < 1$, then the expected true proportion of equilibria is bounded as

$$\mathbb{E}_{\mathcal{P}_1,\ldots,\mathcal{P}_n}[\pi(\mathcal{G})] \leq \kappa^n \ .$$

Furthermore, the following high probability statement

$$\mathbb{P}_{\mathcal{P}_1,\ldots,\mathcal{P}_n}[\pi(\mathcal{G}) \leq \tfrac{\kappa^n}{\delta}] \geq 1 - \delta$$

holds.

Proof Let $f_i(\mathbf{w}_{i,-i}, b_i, \mathbf{x}) \equiv 1[x_i({\mathbf{w}_{i,-i}}^{\mathrm{T}}\mathbf{x}_{-i} - b_i) \geq 0]$ and $\mathcal{P} \equiv \{\mathcal{P}_1, \ldots, \mathcal{P}_n\}$. By Eq. (2), $\mathbb{E}_{\mathcal{P}}[\pi(\mathcal{G})] = \frac{1}{2^n}\sum_{\mathbf{x}} \mathbb{E}_{\mathcal{P}}[\prod_i f_i(\mathbf{w}_{i,-i}, b_i, \mathbf{x})]$. For any $\mathbf{x}$, $f_1(\mathbf{w}_{1,-1}, b_1, \mathbf{x}), \ldots, f_n(\mathbf{w}_{n,-n}, b_n, \mathbf{x})$ are independent since $(\mathbf{w}_{1,-1}, b_1), \ldots, (\mathbf{w}_{n,-n}, b_n)$ are independently distributed. Thus, $\mathbb{E}_{\mathcal{P}}[\pi(\mathcal{G})] = \frac{1}{2^n}\sum_{\mathbf{x}} \prod_i \mathbb{E}_{\mathcal{P}_i}[f_i(\mathbf{w}_{i,-i}, b_i, \mathbf{x})]$. Since for all i and $\mathbf{x}$, $\mathbb{E}_{\mathcal{P}_i}[f_i(\mathbf{w}_{i,-i}, b_i, \mathbf{x})] = \mathbb{P}_{\mathcal{P}_i}[x_i({\mathbf{w}_{i,-i}}^{\mathrm{T}}\mathbf{x}_{-i} - b_i) \geq 0] \leq \kappa$, we have $\mathbb{E}_{\mathcal{P}}[\pi(\mathcal{G})] \leq \kappa^n$.

By Markov's inequality, given that $\pi(\mathcal{G}) \geq 0$, we have $\mathbb{P}_{\mathcal{P}}[\pi(\mathcal{G}) \geq c] \leq \frac{\mathbb{E}_{\mathcal{P}}[\pi(\mathcal{G})]}{c} \leq \frac{\kappa^n}{c}$. For $c = \frac{\kappa^n}{\delta} \Rightarrow \mathbb{P}_{\mathcal{P}}[\pi(\mathcal{G}) \geq \frac{\kappa^n}{\delta}] \leq \delta \Rightarrow \mathbb{P}_{\mathcal{P}}[\pi(\mathcal{G}) \leq \frac{\kappa^n}{\delta}] \geq 1 - \delta$. ■

Remark 28 *Under the same assumptions of Theorem 27, it is possible to prove that with probability at least $1 - \delta$ we have $\pi(\mathcal{G}) \leq \kappa^n + \sqrt{\frac{1}{2}\log\frac{1}{\delta}}$ by using Hoeffding's lemma. We point out that such a bound is not better than the Markov's bound derived above.*

8. Experimental Results

For learning LIGs we used our convex loss methods: independent and simultaneous SVM and logistic regression (See Section 6.5). Additionally, we used the (super-exponential) exhaustive search method (See Section 6.2) only for $n \leq 4$. As a baseline, we used the (NP-hard) sigmoidal maximum likelihood only for $n \leq 15$ as well as the sigmoidal maximum empirical proportion of equilibria (See Section 6.4). Regarding the parameters α and β our sigmoidal function in Eq. (8), we found experimentally that $\alpha = 0.1$ and $\beta = 0.001$ achieved the best results.

For reasons briefly discussed at the end of Section 2.1, we have little interest in determining how much worst game-theoretic models are relative to probabilistic models when applied to data from purely *probabilistic* processes, without any strategic component, as we think this to be a futile exercise. We believe the same is true for evaluating the quality of a probabilistic graphical model vs. a game-theoretic model when applied to *strategic behavioral data*, resulting from a process defined by game-theoretic concepts based on the (stable) outcomes of a game. Nevertheless, we summarize some experiments in Figure 3 that should help illustrate the point of discussed at the end of Section 2.1.

Still, for scientific curiosity, we compare LIGs to learning Ising models. Once again, our goal is not to show the superiority of either games or Ising models. For $n \leq 15$ players, we perform exact ℓ_1-regularized maximum likelihood estimation by using the FOBOS algorithm (Duchi and Singer, 2009a,b) and exact gradients of the log-likelihood of

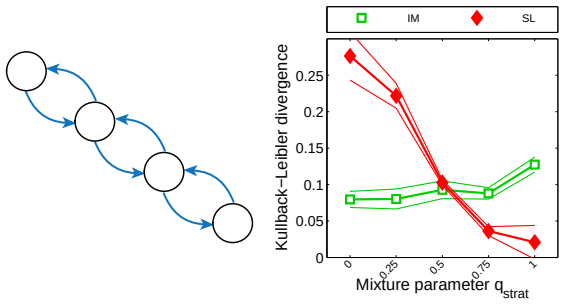


Figure 3: **On the Distinction between Game-Theoretic and Probabilistic Models in the Context of "Probabilistic" vs. "Strategic" Behavioral Data.** The plot shows the performance of Ising models (in green) vs. LIGs (in red) when we learn models from each respective class from data generated by drawing i.i.d. samples from a mixture model of an Ising model, p_{Ising}, and our PSNE-based generative model, p_{LIG}, with mixing parameter q_{strat} corresponding to the probability that the sample is drawn from the LIG component. Hence, we can view q_{strat} as controlling the proportion of the data that is "strategic" in nature. The graph of the Ising model is an (undirected) chain with 4 variable nodes, while that of the LIG is, as shown in the left, also a chain of 4 players with arcs between every consecutive pair of nodes. The parameters of each mixture component in the "ground-truth" mixture model $p_{\text{mix}}(\mathbf{x}) \equiv q_{\text{strat}}\, p_{\text{LIG}}(\mathbf{x}) + (1 - q_{\text{strat}})\, p_{\text{Ising}}(\mathbf{x})$ are the same: node-potential/bias-threshold parameters are all 0; weights of all the edges is $+1$. We set the "signal" parameter q of our generative model p_{LIG} to 0.9. The x-axis of the plot in the right-hand side above corresponds to the mixture parameter q_{strat}; so that, as we move from left to right in the plot, more proportion of the data is "strategic" in nature: $q_{\text{strat}} = 0$ means the data is "purely probabilistic" while $q_{\text{strat}} = 1$ means it is "purely strategic." For each value of $q_{\text{strat}} \in \{0, 0.25, 0.50, 0.75, 1\}$, we generated 50 pairs of data sets from p_{mix}, each of size 50, each pair corresponding to a training and a validation data set, respectively. The learning methods used the validation data set to estimate their respective ℓ_1 regularization parameter. The Ising models learned correspond *exactly* to the optimal penalized likelihood. We use a simultaneous logistic regression approach, described in Section 6, to learn LIGs. In the y-axis of the plot in the right-hand side is the average, over the 50 repetitions, of the *exact* KL-divergence between the respective learned model and $p_{\text{mix}}(\mathbf{x})$. We also include (a linear interpolation of the individual) error bars at 95% confidence level. The plot clearly shows that the more "strategic" the data the better the game-theoretic-based generative model. We can see that the learned Ising models (1) do considerably better than the LIG models when the data is purely probabilistic; and (2) are more "robust" across the spectrum, degrading very gracefully as the data becomes more strategic in nature; but (3) seem to need more data to learn when the data comes exclusively from an Ising model than the LIG model does when the data is purely strategic: The LIG achieves KL values much closer to 0 when the data is purely strategic than the Ising model does when the data is purely probabilistic.

the Ising model. Since the computation of the exact gradient at each step is NP-hard, we used this method only for $n \leq 15$. For $n > 15$ players, we use the Höfling-Tibshirani method (Höfling and Tibshirani, 2009), which uses a sequence of first-order approximations of the exact log-likelihood. We also used a two-step algorithm, by first learning the structure by ℓ_1-regularized logistic regression (Wainwright et al., 2007) and then using the FOBOS algorithm (Duchi and Singer, 2009a,b) with belief propagation for gradient approximation. We did not find a statistically significant difference between the test log-likelihood of both algorithms and therefore we only report the latter.

Our experimental setup is as follows: after learning a model for different values of the regularization parameter ρ in a training set, we select the value of ρ that maximizes the log-likelihood in a validation set, and report statistics in a test set. For synthetic experiments, we report the Kullback-Leibler (KL) divergence, average precision (one minus the fraction of falsely included equilibria), average recall (one minus the fraction of falsely excluded equilibria) in order to measure the closeness of the recovered models to the ground truth. For real-world experiments, we report the log-likelihood. In both synthetic and real-world experiments, we report the number of equilibria and the empirical proportion of equilibria. Our results are statistically significant, we avoided showing error bars for clarity of presentation since error bars and markers overlapped.

8.1 Experiments on Synthetic Data

We first test the ability of the proposed methods to recover the PSNE induced by ground-truth games from data when those games are LIGs. We use a small first synthetic model in order to compare with the (super-exponential) exhaustive search method. The ground-truth model $\mathcal{G}_g = (\mathbf{W}_g, \mathbf{b}_g)$ has $n = 4$ players and 4 Nash equilibria (i.e., $\pi(\mathcal{G}_g)$=0.25), $\mathbf{W}_g$ was set according to Figure 4 (the weight of each edge was set to +1) and $\mathbf{b}_g = \mathbf{0}$. The mixture parameter of the ground-truth model q_g was set to 0.5,0.7,0.9. For each of 50 repetitions, we generated a training, a validation and a test set of 50 samples each. Figure 4 shows that our convex loss methods and sigmoidal maximum likelihood outperform (lower KL) exhaustive search, sigmoidal maximum empirical proportion of equilibria and Ising models. Note that the exhaustive search method which performs exact maximum likelihood suffers from over-fitting and consequently does not produce the lowest KL. From all convex loss methods, simultaneous logistic regression achieves the lowest KL. For all methods, the recovery of equilibria is perfect for $q_g = 0.9$ (number of equilibria equal to the ground truth, equilibrium precision and recall equal to 1). Additionally, the empirical proportion of equilibria resembles the mixture parameter of the ground-truth model q_g.

Next, we use a slightly larger second synthetic model with more complex interactions. We still keep the model small enough in order to compare with the (NP-hard) sigmoidal maximum likelihood method. The ground truth model $\mathcal{G}_g = (\mathbf{W}_g, \mathbf{b}_g)$ has $n = 9$ players and 16 Nash equilibria (i.e., $\pi(\mathcal{G}_g)$=0.03125), $\mathbf{W}_g$ was set according to Figure 5 (the weight of each blue and red edge was set to +1 and −1 respectively) and $\mathbf{b}_g = \mathbf{0}$. The mixture parameter of the ground truth q_g was set to 0.5,0.7,0.9. For each of 50 repetitions, we generated a training, a validation and a test set of 50 samples each. Figure 5 shows that our convex loss methods outperform (lower KL) sigmoidal methods and Ising models. From all convex loss methods, simultaneous logistic regression achieves the lowest KL. For convex

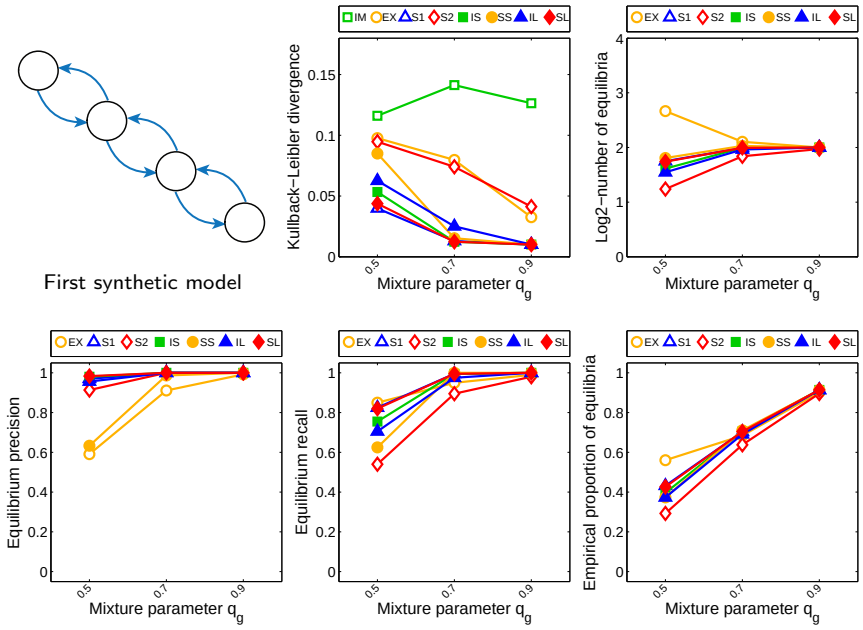


Figure 4: **Closeness of the Recovered Models to the Ground-Truth Synthetic Model for Different Mixture Parameters q_g.** Our convex loss methods (IS,SS: independent and simultaneous SVM, IL,SL: independent and simultaneous logistic regression) and sigmoidal maximum likelihood (S1) have lower KL than exhaustive search (EX), sigmoidal maximum empirical proportion of equilibria (S2) and Ising models (IM). For all methods, the recovery of equilibria is perfect for $q_g = 0.9$ (number of equilibria equal to the ground truth, equilibrium precision and recall equal to 1) and the empirical proportion of equilibria resembles the mixture parameter of the ground truth q_g.

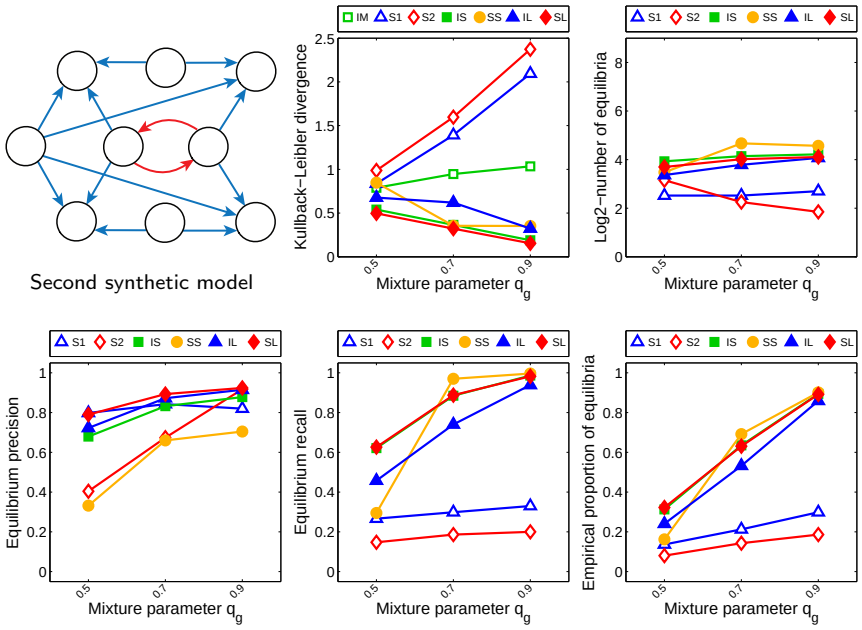


Figure 5: Closeness of the recovered models to the ground truth synthetic model for different mixture parameters q_g. Our convex loss methods (IS,SS: independent and simultaneous SVM, IL,SL: independent and simultaneous logistic regression) have lower KL than sigmoidal maximum likelihood (S1), sigmoidal maximum empirical proportion of equilibria (S2) and Ising models (IM). For convex loss methods, the equilibrium recovery is better than the remaining methods (number of equilibria equal to the ground truth, higher equilibrium precision and recall) and the empirical proportion of equilibria resembles the mixture parameter of the ground truth q_g.

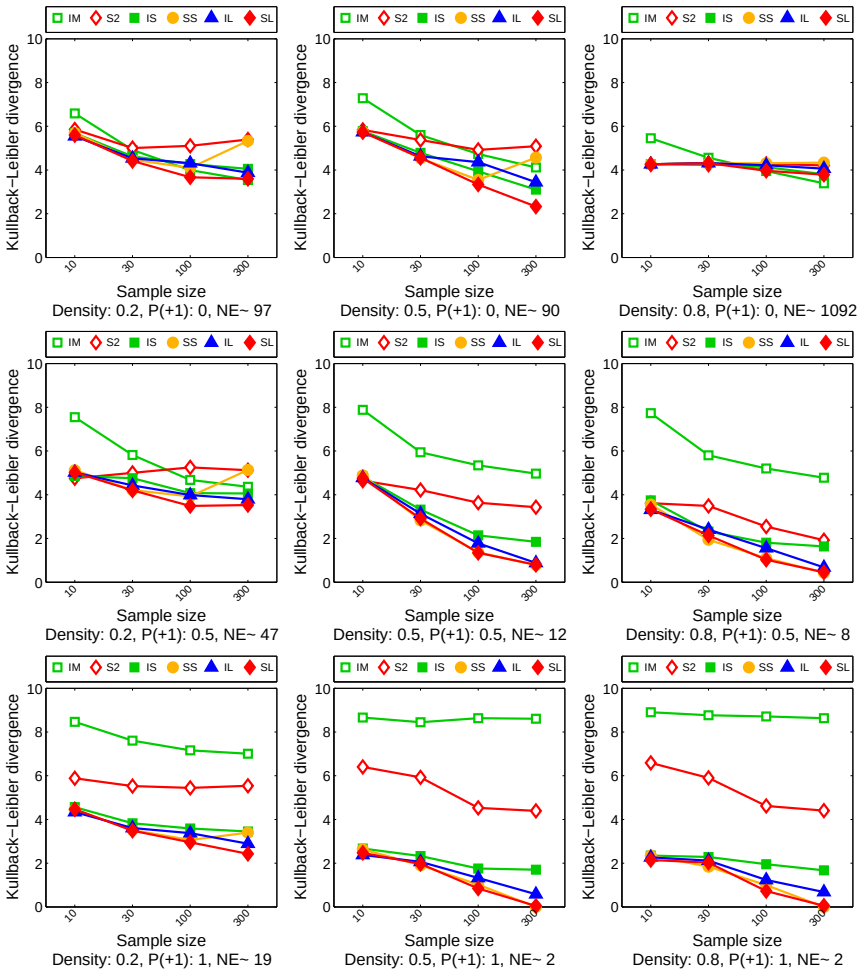


Figure 6: KL divergence between the recovered models and the ground truth for data sets of different number of samples. Each chart shows the density of the ground truth, probability $P(+1)$ that an edge has weight $+1$, and average number of equilibria (NE). Our convex loss methods (IS,SS: independent and simultaneous SVM, IL,SL: independent and simultaneous logistic regression) have lower KL than sigmoidal maximum empirical proportion of equilibria (S2) and Ising models (IM). The results are remarkably better when the number of equilibria in the ground truth model is small (e.g., for NE<20).

loss methods, the equilibrium recovery is better than the remaining methods (number of equilibria equal to the ground truth, higher equilibrium precision and recall). Additionally, the empirical proportion of equilibria resembles the mixture parameter of the ground truth q_g.

In the next experiment, we show that the performance of convex loss minimization improves as the number of samples increases. We used random graphs with slightly more variables and varying number of samples (10,30,100,300). The ground truth model $\mathcal{G}_g =$

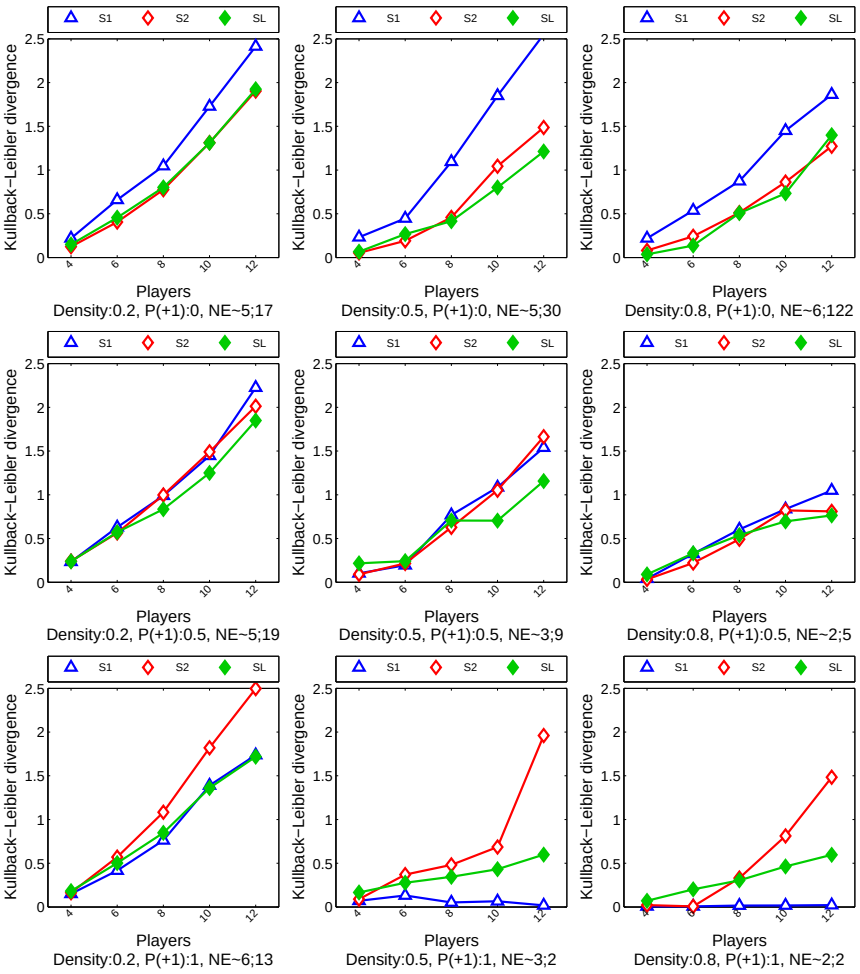


Figure 7: KL divergence between the recovered models and the ground truth for data sets of different number of players. Each chart shows the density of the ground truth, probability $P(+1)$ that an edge has weight $+1$, and average number of equilibria (NE) for $n = 2;n = 14$. In general, simultaneous logistic regression (SL) has lower KL than sigmoidal maximum empirical proportion of equilibria (S2), and the latter one has lower KL than sigmoidal maximum likelihood (S1). Other convex losses behave the same as simultaneous logistic regression (omitted for clarity of presentation).

$(\mathbf{W}_g, \mathbf{b}_g)$ contains $n = 20$ players. For each of 20 repetitions, we generate edges in the ground truth model $\mathbf{W}_g$ with a required density (either 0.2,0.5,0.8). For simplicity, the weight of each edge is set to $+1$ with probability $P(+1)$ and to -1 with probability $1 - P(+1)$.[35] Hence, the Nash equilibria of the generated games does not depend on the magnitude of the weights, just on their sign. We set the bias $\mathbf{b}_g = \mathbf{0}$ and the mixture parameter of the ground truth $q_g = 0.7$. We then generated a training and a validation set with the same number of samples. Figure 6 shows that our convex loss methods outperform (lower KL) sigmoidal maximum empirical proportion of equilibria and Ising models (except for the synthetic model with high true proportion of equilibria: density 0.8, $P(+1) = 0$, NE> 1000). The results are remarkably better when the number of equilibria in the ground truth model is small (e.g., for NE< 20). From all convex loss methods, simultaneous logistic regression achieves the lowest KL.

In the next experiment, we evaluate two effects in our approximation methods. First, we evaluate the impact of removing the true proportion of equilibria from our objective function, i.e., the use of maximum empirical proportion of equilibria instead of maximum likelihood. Second, we evaluate the impact of using convex losses instead of a sigmoidal approximation of the 0/1 loss. We used random graphs with varying number of players and 50 samples. The ground truth model $\mathcal{G}_g = (\mathbf{W}_g, \mathbf{b}_g)$ contains $n = 4, 6, 8, 10, 12$ players. For each of 20 repetitions, we generate edges in the ground truth model $\mathbf{W}_g$ with a required density (either 0.2,0.5,0.8). As in the previous experiment, the weight of each edge is set to $+1$ with probability $P(+1)$ and to -1 with probability $1 - P(+1)$. We set the bias $\mathbf{b}_g = \mathbf{0}$ and the mixture parameter of the ground truth $q_g = 0.7$. We then generated a training and a validation set with the same number of samples. Figure 7 shows that in general, convex loss methods outperform (lower KL) sigmoidal maximum empirical proportion of equilibria, and the latter one outperforms sigmoidal maximum likelihood. A different effect is observed for mild (0.5) to high (0.8) density and $P(+1) = 1$ in which the sigmoidal maximum likelihood obtains the lowest KL. In a closer inspection, we found that the ground truth games usually have only 2 equilibria: $(+1, \ldots, +1)$ and $(-1, \ldots, -1)$, which seems to present a challenge for convex loss methods. It seems that for these specific cases, removing the true proportion of equilibria from the objective function negatively impacts the estimation process, but note that sigmoidal maximum likelihood is not computationally feasible for $n > 15$.

35. Part of the reason for using such "simple"/"limited" binary set of weight values in this synthetic experiment regards the ability to generate "interesting" LIGs; that is, games with interesting sets of PSNE. As a word of caution, this is not as simple as it appears at first glance. LIGs with weights and biases generated *uniformly* at random from some set of real values are almost always not interesting, often having only 1 or 2 PSNE (Irfan and Ortiz, 2014). It is not until we move to more "special"/restricted classes of games, such as that used in this experiments, that more interesting PSNE structure arises from randomly generated LIGs. That is in large part why we concentrated our experiments in games with those simple properties. (Simplicity itself also had a role in our decision, of course.)

    Please understand that we are not saying that LIGs that use a larger set of integers, or non-integer real-valued weights w_{ij}'s or b_i's are not interesting, as the LIGs we learn from the real-world data demonstrate. What we *are* saying is that we do not yet have a good understanding on how to randomly generate "interesting" synthetic games from the standpoint of their induced PSNE. We leave a *comprehensive* evaluation of our MLE-based algorithms' *ability to recover the PSNE of randomly generated synthetic LIGs*, which would involve a diversity of synthetic game graph structures, influence weights and biases that induce "interesting" sets of PSNE, for future work.

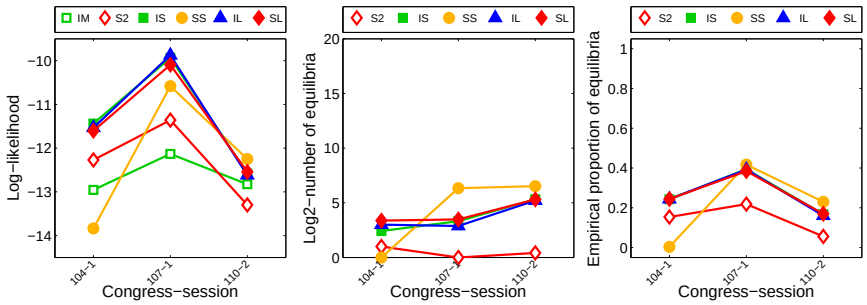


Figure 8: Statistics for games learnt from 20 senators from the first session of the 104th congress, first session of the 107th congress and second session of the 110th congress. The log-likelihood of our convex loss methods (IS,SS: independent and simultaneous SVM, IL,SL: independent and simultaneous logistic regression) is higher than sigmoidal maximum empirical proportion of equilibria (S2) and Ising models (IM). For all methods, the number of equilibria (and so the true proportion of equilibria) is low.

8.2 Experiments on Real-World Data: U.S. Congressional Voting Records

We used the U.S. congressional voting records in order to measure the generalization performance of convex loss minimization in a real-world data set. The data set is publicly available at `http://www.senate.gov/`. We used the first session of the 104th congress (Jan 1995 to Jan 1996, 613 votes), the first session of the 107th congress (Jan 2001 to Dec 2001, 380 votes) and the second session of the 110th congress (Jan 2008 to Jan 2009, 215 votes). Following on other researchers who have experimented with this data set (e.g., Banerjee et al. 2008), abstentions were replaced with negative votes. Since reporting the log-likelihood requires computing the number of equilibria (which is NP-hard), we selected only 20 senators by stratified random sampling. We randomly split the data into three parts. We performed six repetitions by making each third of the data take turns as training, validation and testing sets. Figure 8 shows that our convex loss methods outperform (higher log-likelihood) sigmoidal maximum empirical proportion of equilibria and Ising models. From all convex loss methods, simultaneous logistic regression achieves the lowest KL. For all methods, the number of equilibria (and so the true proportion of equilibria) is low.

We apply convex loss minimization to larger problems, by learning structures of games from all 100 senators. Figure 9 shows that simultaneous logistic regression produce structures that are sparser than its independent counterpart. The simultaneous method better elicits the bipartisan structure of the congress. We define the *(aggregate) direct influence* of player j to all other players as $\sum_i |w_{ij}|$ after normalizing all weights, i.e., for each player i we divide $(\mathbf{w}_{i,-i}, b_i)$ by $\|\mathbf{w}_{i,-i}\|_1 + |b_i|$. Note that Jeffords and Clinton are one of the 5 most directly-influential as well as 5 least directly-influenceable (high bias) senators, in the 107th and 110th congress respectively. McCain and Feingold are both in the list of 5 most directly-influential senators in the 104th and 107th congress. McCain appears again in the list of 5 least *directly* influenceable senators in the 110th congress (as defined above in the context of the LIG model).

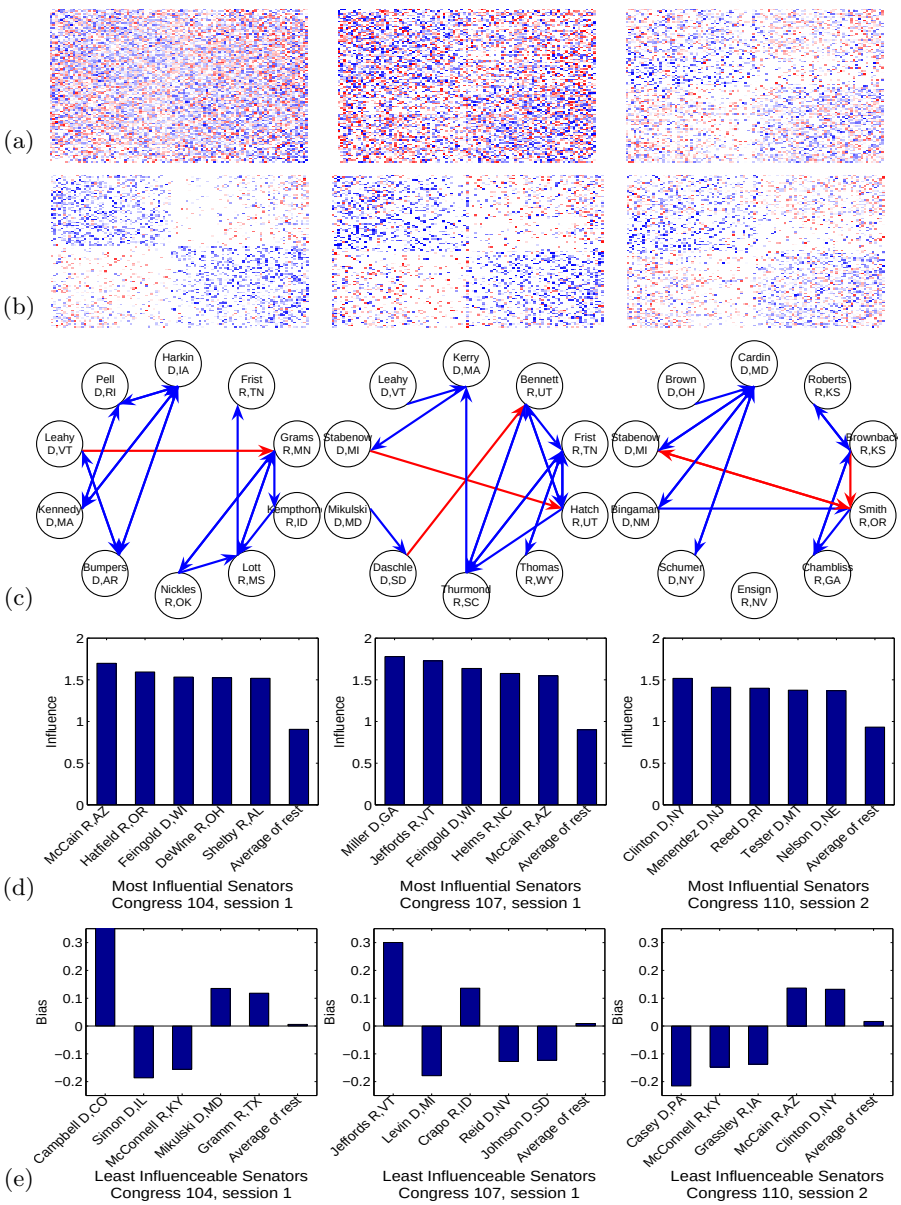


Figure 9: (Top) **Matrices of (direct) influence weights W for games learned from all 100 senators**, from the first session of the 104th congress (left), first session of the 107th congress (center) and second session of the 110th congress (right), by using our independent (a) and simultaneous (b) logistic regression methods. A row represents how much every other senator directly-influence the senator in such row, in terms of the influence weights of the learned LIG. Positive influence-weight parameter values are shown in blue; negative values are in red. Democrats are shown in the top/left corner, while Republicans are shown in the bottom/right corner. Note that simultaneous method produce structures that are sparser than its independent counterpart. (c) **Partial view of the graph for simultaneous logistic regression**. (d) **Most directly-influential senators** and (e) **least directly-influenceable senators**. Regularization parameter $\rho = 0.0006$.

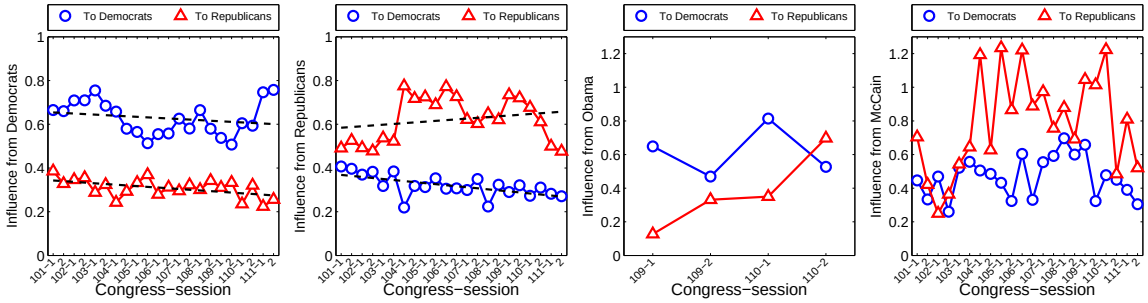


Figure 10: Direct influence between parties and direct influences from Obama and McCain. Games were learned from all 100 senators from the 101th congress (Jan 1989) to the 111th congress (Dec 2010) by using our simultaneous logistic regression method. Direct influence between senators of the same party are stronger than senators of different party, which is also decreasing over time. In the last sessions, direct influence from Obama to Republicans increased, and influence from McCain to both parties decreased. Regularization parameter $\rho = 0.0006$.

We test the hypothesis that the aggregate direct influence, *as defined by our model*, between senators of the same party are stronger than senators of different party. We learn structures of games from all 100 senators from the 101th congress to the 111th congress (Jan 1989 to Dec 2010). The number of votes cast for each session were average: 337, minimum: 215, maximum: 613. Figure 10 validates our hypothesis and more interestingly, it shows that influence between different parties is decreasing over time. Note that the influence from Obama to Republicans increased in the last sessions, while McCain's influence to Republicans decreased.

Since the U.S. Congressional voting data is observational, we used the log-likelihood as an adequate measure of predictive performance. We argue that the log-likelihood of joint actions provides a more "global view" compared to predicting the action of a single agent. Furthermore, predicting the action of a single agent (i.e., x_i) works under the assumption that we have access to the decisions of the other agents (i.e., $\mathbf{x}_{-i}$), which is in contrast to our framework. Regarding causal strategic inference, Irfan and Ortiz (2014) use the games that we produce in this section in order to address problems such as the identification of most influential senators. (We refer the reader to their paper for further details.)

9. Concluding Remarks

In Section 6, we present a variety of algorithms to learn LIGs from strictly behavioral data, including what we call *independent logistic regression (ILR)*. There is a very popular technique for learning Ising models that uses independent regularized logistic regression to compute the individual conditional probabilities *as a step toward computing a globally coherent joint probability distribution*. However, this approach is inherently problematic, as some authors have previously pointed out (see, e.g., Guo et al. 2010). Without getting too technical, the main roadblock is that there is no guarantee that estimates of the weights produced by the individual regressions be symmetric: $\widehat{w}_{ij} = \widehat{w}_{ji}$ for all i, j. Learning an

Ising model requires the enforcement of this condition, and a variety of heuristics have been proposed. (Please see Section 2.1 for relevant work and references in this area.)

We also *apply* ILR in *exactly* the same manner but for a *different objective*: learning LIGs. Some seem to think that this diminishes the significance of our contributions. We strongly believe the opposite is true: That we can learn games by using such simple, practical, efficient and well-studied techniques is a significant plus in our view. Again, without getting too technical, the estimates of ILR need not be symmetric for LIG models, and are always perfectly consistent with the LIG definition. In fact, asymmetric estimates are common in practice (the LIG for the 110th Congress depicted in Figure 1 is an example). And we believe this makes the model more interesting. In the ILR-learned LIG, a player may have a positive, negative or no direct effect on another players utility, and *vice versa*.[36]

Thus, despite the *process* of estimation of model parameters being similar, the *view* of the output of that estimation process is radically different in each case. Our experiments show that our generative model with LIGs built from ILR estimates achieves higher generalization likelihoods than standard probabilistic models such as Ising models that may also use ILR. This fact, that the generative model defined in terms of game-theoretic equilibrium concepts can explain the data better than traditional probabilistic models, provides further evidence supporting such a game-theoretic "view" of the ILR estimated parameters and yields additional confidence in their use in game-theoretic models.

In short, ILR is a thoroughly studied method with a long tradition and an extensive literature from which we can only benefit. We find it to a be a good, unexpected outcome of our research in this work, and thus a reasonably significant contribution, that we can successfully and effectively use ILR, a very simple and practical estimation technique for learning probabilistic graphical models, to learn game-theoretic graphical models too.

9.1 Extensions and Future Work

There are several ways of extending this research. We can extend our approach to ϵ-approximate PSNE.[37] In this case, for each player instead of one condition, we will have two best-response conditions which are still linear in $\mathbf{W}$ and $\mathbf{b}$. Additionally, we can extend our approach to a broader class of graphical games and non-Boolean actions. Note that our analysis does not rely on binary actions, but on binary features of one player $1[x_i = 1]$ or two players $1[x_i = x_j]$. We can use features of three players $1[x_i = x_j = x_k]$ or of non-Boolean actions $1[x_i = 3, x_j = 7]$. This kernelized version is still linear in $\mathbf{W}$ and $\mathbf{b}$. These extensions are possible because our algorithms and analysis rely on linearity and binary features; additionally, we can obtain a new upper-bound on the "VC-dimension" by changing the inputs of the neural-network architecture. We can easily extend our approach to parameter learning for *fixed* structures by using a ℓ_2^2 regularizer instead.

36. It is easy to come up with examples of such opposing/antagonistic interactions between individuals in real-world settings. (See, e.g., "parents with teenagers," perhaps a more timely example is the U.S. Congress in recent times.)

37. By definition, given $\epsilon \geq 0$, a joint pure-strategy $\mathbf{x}^*$ is an *ϵ-approximate PSNE* if for each player i, we have $u_i(\mathbf{x}^*) \geq \max_{x_i} u_i(x_i, \mathbf{x}^*_{-i}) - \epsilon$; in other words, no players can gain more than ϵ in payoff/utility value from unilaterally deviating from x_i^*, assuming the other players play $\mathbf{x}^*_{-i}$. Using this definition, we can see that a PSNE is simply a 0-approximate PSNE.

Future work should also consider and study more sophisticated noise processes, MSNE, and the analysis of different upper bounds for the 0/1 loss (e.g., exponential, smooth hinge). Finally, we should consider other slightly more complex versions of our model based on Bayesian or stochastic games to account for possible variations of the influence-weights and bias-threshold parameters. As an example, we may consider versions of our model for congressional voting that would explicitly capture game differences in terms of influences and biases that depend on the nature or topic of each specific bill being voted on, as well as Senators' time-changing preferences and trends.

Acknowledgements

We are grateful to Christian Luhmann for extensive discussion of many aspects of this work and the multiple comments and feedback he provided to improve both the work and the presentation. We warmly thank Tommi Jaakkola for several informal discussions on the topic of learning games and the ideas behind causal strategic inference, as well as suggestions on how to improve the presentation. We also thank Mohammad Irfan for his help with motivating causal strategic inference in inference games and examples to illustrate their use. We would also like to thank Alexander Berg, Tamara Berg and Dimitris Samaras for their comments and questions during informal presentations of this work, which helped us tailor the presentation to a wider audience. We are very grateful to Nicolle Gruzling for sharing her Master's thesis (Gruzling, 2006) which contains valuable references used in Section 6.2. Finally, we thank several anonymous reviewers for their comments and feedback, and in particular, an anonymous reviewer for the reference to an overview on the literature on composite marginal likelihoods.

Shorter versions of the work presented here appear as Chapter 7 of the first author's Ph.D. dissertation (Honorio, 2012) and as e-print `arXiv:1206.3713 [cs.LG]` (Honorio and Ortiz, 2012).

This work was supported in part by NSF CAREER Award IIS-1054541.

Appendix A. Additional Discussion

In this section, we discuss our choice of modeling end-state predictions without modeling dynamics. We also discuss some alternative noise models to the one studied in this manuscript.

A.1 On End-State Predictions without Explicitly Modeling Dynamics

Certainly, in cases in which information about the dynamics is available, the learned model *may* use such information while still making end-state predictions. But no such information, either via data sequences or prior knowledge, is available in any of the publicly-available real-world data sets we study here. Take congressional voting as an example. Considering the voting records as sequence of votes does not seem sensible in our context from a modeling/engineering perspective because the data set does not have any detailed information about the nature of the vote: we just have each senator's vote on whatever bill they considered, and little to no information about the detailed dynamics that might have lead to the senators' final votes. Indeed, one may go further and argue that assuming the the

availability of information about the dynamics of the process is a considerable burden on the modeler and something of a wishful thinking in many practical, real-world settings.

Besides the nature of our setting, the lack of data or information, and the CSI motivation, there are other more fundamental ML reasons why we have no interest in considering dynamics in this paper. First, we view the additional complexity of a dynamic/temporal model as providing the wrong tradeoff: dynamic/temporal models are often inherently more complex to express and learn from data. Second, it is common ML practice to separate single example/outcome problems from sequence problems; said differently, ML generally treats the problem of learning from individual i.i.d. examples different from that of learning from sequences or sequence prediction. Third, we invoke Vapnik's Principle of Transduction (Vapnik, 1998, page 477):[38]

> "When solving a problem of interest, do not solve a more general problem as an intermediate step. Try to get the answer that you really need but not a more general one."

We believe this additional complexity of temporal dynamics, while possibly more "realistic," might easily weaken the power of the "bottom-line" prediction of the possible stable final outcomes because the resulting models can get side-tracked by the details of the interaction. We believe the difficulty of modeling such details, specially with the relatively limited amount of the data available, leads to poor prediction performance on what we really care about from an *engineering* stand point: We would like to know or predict *what* will end up happening, and have little or no interest on the *how* or *why* this happens.

We recognize the *scientific* significance and importance of research in social and behavioral sciences such as sociology, psychology and, in some cases economics, on explaining, at a higher level of abstraction, often going as low as the "cognitive" or "neuroscience" level, the process by which final decisions are reached.

We believe the sudden growth of interest from industry (both online and physical companies), government and other national and international institutions on predicting "behavior" for the purpose of revenue, improving efficiency, instituting effective policies with minimal regulations, etc., should shift the focus of the study of "behavior" closer to an engineering endeavor. We believe such entities are after the "bottom line" and will care more about the end-goal than *how or why a specific outcome is achieved*, modulo, of course, having simple enough and tractable computational models that provide reasonably accurate predictions of final end-state behavior, or at least accurate enough for their purposes.

A.2 On Alternative Noise Models

Next, we discuss some alternative noise models to the one studied in this manuscript. Specifically, we discuss an extension of the PSNE-based mixture noise model as well as individual-player noise models.

A.2.1 On Generalizations of the PSNE-based Mixture Noise Models

A simple extension to our model in Eq. (1) is to allow for more general distributions for the PSNE and the non-PSNE sets. That is, with some probability $0 < q < 1$, a joint action $\mathbf{x}$ is

38. See also `http://www.cs.man.ac.uk/~jknowles/transductive.html` additional information.

chosen from $\mathcal{NE}(\mathcal{G})$ by following a distribution $\mathcal{P}_\alpha$ parameterized by α; otherwise, $\mathbf{x}$ is chosen from its complement set $\{-1,+1\}^n - \mathcal{NE}(\mathcal{G})$ by following a distribution $\mathcal{P}_\beta$ parameterized by β. The corresponding probability mass function (PMF) over joint-behaviors $\{-1,+1\}^n$ parameterized by $(\mathcal{G}, q, \alpha, \beta)$ is

$$p_{(\mathcal{G},q,\alpha,\beta)}(\mathbf{x}) = q \frac{p_\alpha(\mathbf{x}) 1[\mathbf{x} \in \mathcal{NE}(\mathcal{G})]}{\sum_{\mathbf{z} \in \mathcal{NE}(\mathcal{G})} p_\alpha(\mathbf{z})} + (1-q) \frac{p_\beta(\mathbf{x}) 1[\mathbf{x} \notin \mathcal{NE}(\mathcal{G})]}{\sum_{\mathbf{z} \notin \mathcal{NE}(\mathcal{G})} p_\beta(\mathbf{z})} ,$$

where $\mathcal{G}$ is a game, $p_\alpha(\mathbf{x})$ and $p_\beta(\mathbf{x})$ are PMFs over $\{-1,+1\}^n$.

One reasonable technique to learn such a model from data is to perform an alternate method. Compared to our simpler model, this model requires a step that maximizes the likelihood by changing α and β while keeping $\mathcal{G}$ and q constant. The complexity of this step will depend on how we parameterize the PMFs $\mathcal{P}_\alpha$ and $\mathcal{P}_\beta$ but will very likely be an NP-hard problem because of the partition function. Furthermore, the problem of maximizing the likelihood by changing $\mathcal{NE}(\mathcal{G})$ (while keeping α, β and q constant) is combinatorial in nature and in this paper, we provided a tractable approximation method with provable guarantees (for the case of uniform distributions). Other approaches for maximizing the likelihood by changing $\mathcal{G}$ are very likely to be exponential as we discuss briefly at the start of Section 6.

A.2.2 On Individual-Player Noise Models

As an example, consider a the generative model in which we first randomly select a PSNE $\mathbf{x}$ of the game from a distribution $\mathcal{P}_\alpha$ parameterized by α, and then each player i, independently, acts according to x_i with probability q_i and switches its action with probability $1-q_i$. The corresponding probability mass function (PMF) over joint-behaviors $\{-1,+1\}^n$ parameterized by $(\mathcal{G}, q_1, \ldots, q_n, \alpha)$ is

$$p_{\mathcal{G},\mathbf{q},\alpha}(\mathbf{x}) = \sum_{\mathbf{y} \in \mathcal{NE}(\mathcal{G})} \frac{p_\alpha(\mathbf{y})}{\sum_{\mathbf{z} \in \mathcal{NE}(\mathcal{G})} p_\alpha(\mathbf{z})} \prod_i q_i^{1[x_i = y_i]} (1-q_i)^{1[x_i \neq y_i]} ,$$

where $\mathcal{G}$ is a game, $\mathbf{q} = (q_1, \ldots, q_n)$ and $p_\alpha(\mathbf{y})$ is a PMF over $\{-1,+1\}^n$.

One reasonable technique to learn such a model from data is to perform an alternate method. In one step, we maximize the likelihood by changing $\mathcal{NE}(\mathcal{G})$ while keeping $\mathbf{q}$ and α constant, which could be tractably performed by applying Jensen's inequality since the problem is combinatorial in nature. In another step, we maximize the likelihood by changing $\mathbf{q}$ while keeping $\mathcal{G}$ and α constant, which could be performed by gradient ascent (the complexity of this step depends on the size of $\mathcal{NE}(\mathcal{G})$ which could be exponential!). In yet another step, we maximize the likelihood by changing α while keeping $\mathcal{G}$ and $\mathbf{q}$ constant (the complexity of this step will depend on how we parameterize the PMF $\mathcal{P}_\alpha$ but will very likely be an NP-hard problem because of the partition function). The main problem with this technique is that it can be formally proved that the first step (Jensen's inequality) will almost surely pick a single equilibria for the model (i.e., $|\mathcal{NE}(\mathcal{G})| = 1$).

References

H. Ackermann and A. Skopalik. On the complexity of pure Nash equilibria in player-specific network congestion games. In X. Deng and F. Graham, editors, *Internet and Network*

Economics, volume 4858 of *Lecture Notes in Computer Science*, pages 419–430. Springer Berlin Heidelberg, 2007.

O. Aichholzer and F. Aurenhammer. Classifying hyperplanes in hypercubes. *SIAM Journal on Discrete Mathematics*, 9(2):225–232, 1996.

R. Aumann. Subjectivity and correlation in randomized strategies. *Journal of Mathematical Economics*, 1:67–96, 1974.

C. Ballester, A. Calvó-Armengol, and Y. Zenou. Who's who in crime networks. wanted: the key player. Working Paper Series 617, Research Institute of Industrial Economics, March 2004.

C. Ballester, A. Calvó-Armengol, and Y. Zenou. Who's who in networks. wanted: The key player. *Econometrica*, 74(5):1403–1417, 2006.

O. Banerjee, L. El Ghaoui, A. d'Aspremont, and G. Natsoulis. Convex optimization techniques for fitting sparse Gaussian graphical models. In W. Cohen and A. Moore, editors, *Proceedings of the 23nd International Machine Learning Conference*, pages 89–96. Omni Press, 2006.

O. Banerjee, L. El Ghaoui, and A. d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *Journal of Machine Learning Research*, 9:485–516, 2008.

B. Blum, C.R. Shelton, and D. Koller. A continuation method for Nash equilibria in structured games. *Journal of Artificial Intelligence Research*, 25:457–502, 2006.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2006.

P. S. Bradley and O. L. Mangasarian. Feature selection via concave minimization and support vector machines. In *Proceedings of the 15th International Conference on Machine Learning*, ICML '98, pages 82–90, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

W. Brock and S. Durlauf. Discrete choice with social interactions. *The Review of Economic Studies*, 68(2):235–260, 2001.

C. Camerer. *Behavioral Game Theory: Experiments on Strategic Interaction*. Princeton University Press, 2003.

T. Cao, X. Wu, T. Hu, and S. Wang. Active learning of model parameters for influence maximization. In D. Gunopulos, T. Hofmann, D. Malerba, and M. Vazirgiannis, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 6911 of *Lecture Notes in Computer Science*, pages 280–295. Springer Berlin Heidelberg, 2011.

A. Chapman, A. Farinelli, E. Munoz de Cote, A. Rogers, and N. Jennings. A distributed algorithm for optimising over pure strategy Nash equilibria. In *AAAI Conference on Artificial Intelligence*, 2010.

D. Chickering. Learning equivalence classes of Bayesian-network structures. *Journal of Machine Learning Research*, 2:445–498, 2002.

C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467, 1968.

C. Daskalakis and C.H. Papadimitriou. Computing pure Nash equilibria in graphical games via Markov random fields. In *Proceedings of the 7th ACM Conference on Electronic Commerce*, EC '06, pages 91–99, New York, NY, USA, 2006. ACM.

C. Daskalakis, P. Goldberg, and C. Papadimitriou. The complexity of computing a Nash equilibrium. *Communications of the ACM*, 52(2):89–97, 2009.

C. Daskalakis, A. Dimakisy, and E. Mossel. Connectivity and equilibrium in random games. *Annals of Applied Probability*, 21(3):987–1016, 2011.

B. Dilkina, C. P. Gomes, and A. Sabharwal. The impact of network topology on pure Nash equilibria in graphical games. In *Proceedings of the 22nd National Conference on Artificial Intelligence - Volume 1*, AAAI'07, pages 42–49. AAAI Press, 2007.

P. Domingos. Mining social networks for viral marketing. *IEEE Intelligent Systems*, 20(1): 80–82, 2005. Short Paper.

P. Domingos and M. Richardson. Mining the network value of customers. In *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '01, pages 57–66, New York, NY, USA, 2001. ACM.

J. Duchi and Y. Singer. Efficient learning using forward-backward splitting. In Y. Bengio, D. Schuurmans, J.D. Lafferty, C.K.I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 495–503. Curran Associates, Inc., 2009a.

J. Duchi and Y. Singer. Efficient online and batch learning using forward backward splitting. *Journal of Machine Learning Research*, 10:2899–2934, 2009b.

J. Dunkel. Complexity of pure-strategy Nash equilibria in non-cooperative games. In K.-H. Waldmann and U. M. Stocker, editors, *Operations Research Proceedings*, volume 2006, pages 45–51. Springer Berlin Heidelberg, 2007.

J. Dunkel and A. Schulz. On the complexity of pure-strategy Nash equilibria in congestion and local-effect games. In P. Spirakis, M. Mavronicolas, and S. Kontogiannis, editors, *Internet and Network Economics*, volume 4286 of *Lecture Notes in Computer Science*, pages 62–73. Springer Berlin Heidelberg, 2006.

Q. Duong, M. Wellman, and S. Singh. Knowledge combination in graphical multiagent model. In *Proceedings of the 24th Annual Conference on Uncertainty in Artificial Intelligence (UAI-08)*, pages 145–152, Corvallis, Oregon, 2008. AUAI Press.

Q. Duong, Y. Vorobeychik, S. Singh, and M.P. Wellman. Learning graphical game models. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, IJCAI'09, pages 116–121, San Francisco, CA, USA, 2009. Morgan Kaufmann Publishers Inc.

Q. Duong, M.P. Wellman, S. Singh, and Y. Vorobeychik. History-dependent graphical multiagent models. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: Volume 1*, AAMAS '10, pages 1215–1222, Richland, SC, 2010. International Foundation for Autonomous Agents and Multiagent Systems.

Q. Duong, M.P. Wellman, S. Singh, and M. Kearns. Learning and predicting dynamic networked behavior with graphical multiagent models. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '12, pages 441–448, Richland, SC, 2012. International Foundation for Autonomous Agents and Multiagent Systems.

E. Even-Dar and A. Shapira. A note on maximizing the spread of influence in social networks. In X. Deng and F. Graham, editors, *Internet and Network Economics*, volume 4858 of *Lecture Notes in Computer Science*, pages 281–286. Springer Berlin Heidelberg, 2007.

A. Fabrikant, C. Papadimitriou, and K. Talwar. The complexity of pure Nash equilibria. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, STOC '04, pages 604–612, New York, NY, USA, 2004. ACM.

S. Ficici, D. Parkes, and A. Pfeffer. Learning and solving many-player games through a cluster-based representation. In *Proceedings of the 24th Annual Conference on Uncertainty in Artificial Intelligence (UAI-08)*, pages 188–195, Corvallis, Oregon, 2008. AUAI Press.

D. Fudenberg and D. Levine. *The Theory of Learning in Games*. MIT Press, 1999.

D. Fudenberg and J. Tirole. *Game Theory*. The MIT Press, 1991.

X. Gao and A. Pfeffer. Learning game representations from data using rationality constraints. In *Proceedings of the 26th Annual Conference on Uncertainty in Artificial Intelligence (UAI-10)*, pages 185–192, Corvallis, Oregon, 2010. AUAI Press.

I. Gilboa and E. Zemel. Nash and correlated equilibria: some complexity considerations. *Games and Economic Behavior*, 1(1):80–93, 1989.

M. Gomez Rodriguez, J. Leskovec, and A. Krause. Inferring networks of diffusion and influence. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 1019–1028, New York, NY, USA, 2010. ACM.

G. Gottlob, G. Greco, and F. Scarcello. Pure Nash equilibria: Hard and easy games. *Journal of Artificial Intelligence Research*, 24(1):357–406, 2005.

A. Goyal, F. Bonchi, and L.V.S. Lakshmanan. Learning influence probabilities in social networks. In *Proceedings of the 3rd ACM International Conference on Web Search and Data Mining*, WSDM '10, pages 241–250, New York, NY, USA, 2010. ACM.

M. Granovetter. Threshold models of collective behavior. *The American Journal of Sociology*, 83(6):1420–1443, 1978.

N. Gruzling. Linear separability of the vertices of an n-dimensional hypercube. Master's thesis, The University of British Columbia, December 2006.

J. Guo, E. Levina, G. Michailidis, and J. Zhu. Joint structure estimation for categorical Markov networks. Technical report, University of Michigan, Department of Statistics, 2010. Submitted. `http://www.stat.lsa.umich.edu/~elevina`.

Y. Guo and D. Schuurmans. Convex structure learning for Bayesian networks: Polynomial feature selection and approximate ordering. In *Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence (UAI-06)*, pages 208–216, Arlington, Virginia, 2006. AUAI Press.

E. Hasan and F. Galiana. Electricity markets cleared by merit order–part II: Strategic offers and market power. *IEEE Transactions on Power Systems*, 23(2):372–379, 2008.

E. Hasan and F. Galiana. Fast computation of pure strategy Nash equilibria in electricity markets cleared by merit order. *IEEE Transactions on Power Systems*, 25(2):722–728, 2010.

E. Hasan, F. Galiana, and A. Conejo. Electricity markets cleared by merit order–part I: Finding the market outcomes supported by pure strategy Nash equilibria. *IEEE Transactions on Power Systems*, 23(2):361–371, 2008.

G. Heal and H. Kunreuther. You only die once: Managing discrete interdependent risks. Working Paper W9885, National Bureau of Economic Research, August 2003.

G. Heal and H. Kunreuther. Supermodularity and tipping. Working Paper 12281, National Bureau of Economic Research, June 2006.

G. Heal and H. Kunreuther. Modeling interdependent risks. *Risk Analysis*, 27:621–634, 2007.

H. Höfling and R. Tibshirani. Estimation of sparse binary pairwise Markov networks using pseudo-likelihoods. *Journal of Machine Learning Research*, 10:883–906, 2009.

J. Honorio. *Tractable Learning of Graphical Model Structures from Data*. PhD thesis, Stony Brook University, Department of Computer Science, August 2012.

J. Honorio and L. Ortiz. Learning the structure and parameters of large-population graphical games from behavioral data. *Computer Research Repository*, 2012. `http://arxiv.org/abs/1206.3713`.

M. Irfan and L. E. Ortiz. On influence, stable behavior, and the most influential individuals in networks: A game-theoretic approach. *Artificial Intelligence*, 215:79–119, 2014.

E. Janovskaja. Equilibrium situations in multi-matrix games. *Litovskiĭ Matematicheskiĭ Sbornik*, 8:381–384, 1968.

A. Jiang and K. Leyton-Brown. Action-graph games. Technical Report TR-2008-13, University of British Columbia, Department of Computer Science, September 2008.

A.X. Jiang and K. Leyton-Brown. Polynomial-time computation of exact correlated equilibrium in compact games. In *Proceedings of the 12th ACM Conference on Electronic Commerce*, EC '11, pages 119–126, New York, NY, USA, 2011. ACM.

S. Kakade, M. Kearns, J. Langford, and L. Ortiz. Correlated equilibria in graphical games. In *Proceedings of the 4th ACM Conference on Electronic Commerce*, EC '03, pages 42–47, New York, NY, USA, 2003. ACM.

M. Kearns. Economics, computer science, and policy. *Issues in Science and Technology*, Winter 2005.

M. Kearns and U. Vazirani. *An Introduction to Computational Learning Theory*. The MIT Press, 1994.

M. Kearns and J. Wortman. Learning from collective behavior. In R.A. Servedio and T. Zhang, editors, *21st Annual Conference on Learning Theory - COLT 2008, Helsinki, Finland, July 9-12, 2008*, COLT '08, pages 99–110. Omnipress, 2008.

M. Kearns, M. Littman, and S. Singh. Graphical models for game theory. In *Proceedings of the 17th Annual Conference on Uncertainty in Artificial Intelligence (UAI-01)*, pages 253–260, San Francisco, CA, 2001. Morgan Kaufmann.

J. Kleinberg. Cascading behavior in networks: Algorithmic and economic issues. In N. Nisan, T. Roughgarden, É. Tardos, and V. V. Vazirani, editors, *Algorithmic Game Theory*, chapter 24, pages 613–632. Cambridge University Press, 2007.

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009.

D. Koller and B. Milch. Multi-agent influence diagrams for representing and solving games. *Games and Economic Behavior*, 45(1):181–221, 2003.

H. Kunreuther and E. Michel-Kerjan. Assessing, managing and benefiting from global interdependent risks: The case of terrorism and natural disasters, August 2007. CREATE Symposium: `http://opim.wharton.upenn.edu/risk/library/AssessingRisks-2007.pdf`.

P. La Mura. Game networks. In *Proceedings of the 16th Annual Conference on Uncertainty in Artificial Intelligence (UAI-00)*, pages 335–342, San Francisco, CA, 2000. Morgan Kaufmann.

S. Lee, V. Ganapathi, and D. Koller. Efficient structure learning of Markov networks using L_1-regularization. In B. Schölkopf, J.C. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 817–824. MIT Press, 2007.

D. López-Pintado and D. Watts. Social influence, binary decisions and collective dynamics. *Rationality and Society*, 20(4):399–443, 2008.

R. McKelvey and T. Palfrey. Quantal response equilibria for normal form games. *Games and Economic Behavior*, 10(1):6–38, 1995.

S. Morris. Contagion. *The Review of Economic Studies*, 67(1):57–78, 2000.

S. Muroga. Lower bounds on the number of threshold functions and a maximum weight. *IEEE Transactions on Electronic Computers*, 14:136–148, 1965.

S. Muroga. *Threshold Logic and Its Applications*. John Wiley & Sons, 1971.

S. Muroga and I. Toda. Lower bound of the number of threshold functions. *IEEE Transactions on Electronic Computers*, 5:805–806, 1966.

J. Nash. Non-cooperative games. *Annals of Mathematics*, 54(2):286–295, 1951.

A. Y. Ng and S. J. Russell. Algorithms for inverse reinforcement learning. In *Proceedings of the 17th International Conference on Machine Learning*, ICML '00, pages 663–670, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

N. Nisan, T. Roughgarden, É. Tardos, and V. V. Vazirani, editors. *Algorithmic Game Theory*. Cambridge University Press, 2007.

L. E. Ortiz and M. Kearns. Nash propagation for loopy graphical games. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 817–824. MIT Press, 2003.

C. Papadimitriou and T. Roughgarden. Computing correlated equilibria in multi-player games. *Journal of the ACM*, 55(3):1–29, 2008.

Y. Rinott and M. Scarsini. On the number of pure strategy Nash equilibria in random games. *Games and Economic Behavior*, 33(2):274–293, 2000.

R. Rosenthal. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory*, 2(1):65–67, 1973.

C.T. Ryan, A.X. Jiang, and K. Leyton-Brown. Computing pure strategy Nash equilibria in compact symmetric games. In *Proceedings of the 11th ACM Conference on Electronic Commerce*, EC '10, pages 63–72, New York, NY, USA, 2010. ACM.

K. Saito, R. Nakano, and M. Kimura. Prediction of information diffusion probabilities for independent cascade model. In I. Lovrek, R. J. Howlett, and L. C. Jain, editors, *Knowledge-Based Intelligent Information and Engineering Systems*, volume 5179 of *Lecture Notes in Computer Science*, pages 67–75. Springer Berlin Heidelberg, 2008.

K. Saito, M. Kimura, K. Ohara, and H. Motoda. Learning continuous-time information diffusion model for social behavioral data analysis. In Z. Zhou and T. Washio, editors, *Advances in Machine Learning*, volume 5828 of *Lecture Notes in Computer Science*, pages 322–337. Springer Berlin Heidelberg, 2009.

K. Saito, M. Kimura, K. Ohara, and H. Motoda. Selecting information diffusion models over social networks for behavioral analysis. In J. L. Balcázar, F. Bonchi, A. Gionis, and M. Sebag, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 6323 of *Lecture Notes in Computer Science*, pages 180–195. Springer Berlin Heidelberg, 2010.

M. Schmidt and K. Murphy. Modeling discrete interventional data using directed cyclic graphical models. In *Proceedings of the 25th Annual Conference on Uncertainty in Artificial Intelligence (UAI-09)*, pages 487–495, Corvallis, Oregon, 2009. AUAI Press.

M. Schmidt, G. Fung, and R. Rosales. Fast optimization methods for l1 regularization: A comparative study and two new approaches. In *Proceedings of the 18th European Conference on Machine Learning*, ECML '07, pages 286–297, Berlin, Heidelberg, 2007a. Springer-Verlag.

M. Schmidt, A. Niculescu-Mizil, and K. Murphy. Learning graphical model structure using ℓ_1-regularization paths. In *Proceedings of the 22nd National Conference on Artificial Intelligence - Volume 2*, pages 1278–1283, 2007b.

Y. Shoham. Computer science and game theory. *Communications of the ACM*, 51(8):74–79, 2008.

Y. Shoham and K. Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, 2009.

E. Sontag. VC dimension of neural networks. *Neural Networks and Machine Learning*, pages 69–95, 1998.

N. Srebro. Maximum likelihood bounded tree-width Markov networks. In *Proceedings of the 17th Annual Conference on Uncertainty in Artificial Intelligence (UAI-01)*, pages 504–511, San Francisco, CA, 2001. Morgan Kaufmann.

W. Stanford. A note on the probability of k pure Nash equilibria in matrix games. *Games and Economic Behavior*, 9(2):238–246, 1995.

A. Sureka and P.R. Wurman. Using tabu best-response search to find pure strategy Nash equilibria in normal form games. In *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '05, pages 1023–1029, New York, NY, USA, 2005. ACM.

V. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

D. Vickrey and D. Koller. Multi-agent algorithms for solving graphical games. In *18th National Conference on Artificial Intelligence*, pages 345–351, Menlo Park, CA, USA, 2002. American Association for Artificial Intelligence.

Y. Vorobeychik, M. Wellman, and S. Singh. Learning payoff functions in infinite games. *Machine Learning*, 67(1-2):145–168, 2007.

M.J. Wainwright, J.D. Lafferty, and P.K. Ravikumar. High-dimensional graphical model selection using ℓ_1-regularized logistic regression. In B. Schölkopf, J.C. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 1465–1472. MIT Press, 2007.

K. Waugh, B. Ziebart, and D. Bagnell. Computational rationalization: The inverse equilibrium problem. In Lise Getoor and Tobias Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ICML '11, pages 1169–1176, New York, NY, USA, June 2011. ACM.

R. Winder. Single state threshold logic. *Switching Circuit Theory and Logical Design*, S-134: 321–332, 1960.

J. Wright and K. Leyton-Brown. Beyond equilibrium: Predicting human behavior in normal-form games. In *AAAI Conference on Artificial Intelligence*, 2010.

J.R. Wright and K. Leyton-Brown. Behavioral game theoretic models: A Bayesian framework for parameter analysis. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '12, pages 921–930, Richland, SC, 2012. International Foundation for Autonomous Agents and Multiagent Systems.

S. Yamija and T. Ibaraki. A lower bound of the number of threshold functions. *IEEE Transactions on Electronic Computers*, 14:926–929, 1965.

J. Zhu, S. Rosset, R. Tibshirani, and T.J. Hastie. 1-norm support vector machines. In S. Thrun, L.K. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, pages 49–56. MIT Press, 2004.

B. Ziebart, J. A. Bagnell, and A. K. Dey. Modeling interaction via the principle of maximum causal entropy. In Johannes Fürnkranz and Thorsten Joachims, editors, *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 1255–1262, Haifa, Israel, June 2010. Omnipress.

Journal of Machine Learning Research 16 (2015) 1211-1242 Submitted 1/14; Revised 7/14; Published 6/15

Local Identification of Overcomplete Dictionaries

Karin Schnass KARIN.SCHNASS@UIBK.AC.AT
Department of Mathematics
University of Innsbruck
Technikerstraße 13
6020 Innsbruck, Austria

Editor: Shie Mannor

Abstract

This paper presents the first theoretical results showing that stable identification of overcomplete μ-coherent dictionaries $\Phi \in \mathbb{R}^{d\times K}$ is locally possible from training signals with sparsity levels S up to the order $O(\mu^{-2})$ and signal to noise ratios up to $O(\sqrt{d})$. In particular the dictionary is recoverable as the local maximum of a new maximization criterion that generalizes the K-means criterion. For this maximization criterion results for asymptotic exact recovery for sparsity levels up to $O(\mu^{-1})$ and stable recovery for sparsity levels up to $O(\mu^{-2})$ as well as signal to noise ratios up to $O(\sqrt{d})$ are provided. These asymptotic results translate to finite sample size recovery results with high probability as long as the sample size N scales as $O(K^3 dS\tilde{\varepsilon}^{-2})$, where the recovery precision $\tilde{\varepsilon}$ can go down to the asymptotically achievable precision. Further, to actually find the local maxima of the new criterion, a very simple Iterative Thresholding and K (signed) Means algorithm (ITKM), which has complexity $O(dKN)$ in each iteration, is presented and its local efficiency is demonstrated in several experiments.

Keywords: dictionary learning, dictionary identification, sparse coding, sparse component analysis, vector quantization, K-means, finite sample size, sample complexity, maximization criterion, sparse representation

1. Introduction

Be it the 300 million photos uploaded to Facebook per day, the 800GB the large Hadron collider records per second or the 320.000GB per second it cannot record, it is clear that we have reached the age of big data. Indeed, in 2012, the amount of data existing worldwide is estimated to have reached 2.8 ZB = 2.800 billion GB and while 23 % of these data are expected to be useful if analyzed, only 1% actually are. So how do we deal with this big data challenge? The key concept, that has driven data processing and data analysis in the past decade, is that even high-dimensional data has intrinsically low complexity, meaning that every data point y can be represented as linear combination of a sparse (small) number of elements or atoms $\phi_i \in \mathbb{R}^d$, $\|\phi\|_2 = 1$ of an overcomplete dictionary $\Phi = (\phi_1, \dots \phi_K)$, that is,

$$y \approx \Phi_I x_I = \sum_{i\in I} x(i)\varphi_i,$$

for a set I of size S, $|I| = S$, which is small compared to the ambient dimension, $S \ll d \leq K$. These sparse components do not only describe the data but the representations can also be

used for a myriad of efficient sparsity based data processing schemes, ranging from denoising (Donoho et al., 2006) to compressed sensing (Donoho, 2006; Candès et al., 2006). Therefore a promising tool both for data analysis and data processing, that has emerged in the last years, is dictionary learning, also known as sparse coding or sparse component analysis. Dictionary learning addresses the fundamental question of how to automatically learn a dictionary, providing sparse representations for a given data class. That is, given N signals $y_n \in \mathbb{R}^d$, stored as columns in a matrix $Y = (y_1, \ldots, y_N)$, find a decomposition

$$Y \approx \Phi X$$

into a $d \times K$ dictionary matrix Φ with unit norm columns and a $K \times N$ coefficient matrix with sparse columns.

Until recently the main research focus in dictionary learning has been on the development of algorithms. Thus by now there is an ample choice of learning algorithms, that perform well in experiments and are popular in applications (Field and Olshausen, 1996; Kreutz-Delgado and Rao, 2000; Kreutz-Delgado et al., 2003; Aharon et al., 2006; Yaghoobi et al., 2009; Mairal et al., 2010; Skretting and Engan, 2010; Rubinstein et al., 2010). However, slowly the interest is shifting and researchers are starting to investigate also the theoretical aspects of dictionary learning. Following the first theoretical insights, originating in the blind source separation community (Zibulevsky and Pearlmutter, 2001; Georgiev et al., 2005), there is now a set of generalization bounds predicting how well a learned dictionary can be expected to sparsely approximate future data (Maurer and Pontil, 2010; Vainsencher et al., 2011; Mehta and Gray, 2012; Gribonval et al., 2013). These results give a theoretical foundation for dictionary learning as data processing tool, for example for compression, but unfortunately do not give guarantees that an efficient algorithm will find/recover a good dictionary provided that it exists. However, in order to justify the use of dictionary learning as data analysis tool, for instance in blind source separation, it is important to provide conditions under which an algorithm or scheme can identify the dictionary from a finite number of training signals, that is, the sources from the mixtures. Following the first dictionary identification results for the ℓ_1-minimization principle (Gribonval and Schnass, 2010; Geng et al., 2011; Jenatton et al., 2014), which was suggested by Zibulevsky and Pearlmutter (2001)/Plumbley (2007), and for the ER-SPuD algorithm for learning a basis (Spielman et al., 2012), 2013 has seen a number of interesting developments. First it was shown that the K-SVD minimization principle suggested by Aharon et al. (2006) can locally identify overcomplete tight dictionaries (Schnass, 2014). Later algorithms with *global* identification guarantees for coherent dictionaries were presented (Arora et al., 2014; Agarwal et al., 2014b). Finally it was shown that an alternating minimization method is locally convergent to the correct generating dictionary (Agarwal et al., 2014a). One aspect that all these results have in common is that the sparsity level of the training signals required for successful identification is of order $O(\mu^{-1})$ or $O(\sqrt{d})$ for incoherent dictionaries. Considering that on average sparse recovery in a given dictionary is successful for sparsity levels $O(\mu^{-2})$ (Tropp, 2008; Schnass and Vandergheynst, 2007) and that for dictionary learning we usually have a lot of training signals at our disposal, the same sparsity level should be sufficient for dictionary learning and indeed in this paper we provide the first indication that global dictionary identification could be possible for sparsity levels $O(\mu^{-2})$ by proving that it is locally possible. Further we show that in experiments a very simple iterative

algorithm, based on thresholding and K signed means, is locally successful.
The paper is organized as follows. After introducing all necessary notation in Section 2 we present a new optimization criterion, motivated by the analysis of the K-SVD principle (Schnass, 2014) in Section 3. In Section 4 we give asymptotic identification results both for exact and stable recovery, which in Section 5 are extended to results to finite sample sizes. Section 6 provides an algorithm for actually finding a local optimum and some experiments confirming the theoretical results. Finally in the last section we compare the results for the new criterion to existing identification results, discuss the implications of these local results for global dictionary identification algorithms and point out directions for future research.

2. Notations and Conventions

Before we jump into the fray, we collect some definitions and lose a few words on notations; usually subscripted letters will denote vectors with the exception of c and ε where they are numbers, e.g., $(x_1, \dots, x_K) = X \in \mathbb{R}^{d\times K}$ vs. $c = (c_1, \dots, c_K) \in \mathbb{R}^K$, however, it should always be clear from the context what we are dealing with.
We consider a **dictionary** Φ a collection of K unit norm vectors $\phi_i \in \mathbb{R}^d$, $\|\phi_i\|_2 = 1$. By abuse of notation we will also refer to the $d \times K$ matrix collecting the atoms as its columns as the dictionary, that is $\Phi = (\phi_i, \dots \phi_K)$. The maximal absolute inner product between two different atoms is called the **coherence** μ of the dictionary, $\mu = \max_{i\neq j} |\langle \phi_i, \phi_j \rangle|$. By Φ_I we denote the restriction of the dictionary to the atoms indexed by I, that is $\Phi_I = (\phi_{i_1} \dots \phi_{i_S})$, $i_j \in I$. We indicate the conjugate transpose of a matrix with a *, for example $\Phi^\star$ would be the transpose of Φ.
The set of all dictionaries of a given size $(d \times K)$ is denoted by $\mathcal{D}$. For two dictionaries $\Phi, \Psi \in \mathcal{D}$ we define the distance between each other as the maximal distance between two corresponding atoms,

$$d(\Phi, \Psi) := \max_i \|\phi_i - \psi_i\|_2.$$

We consider a **frame** F a collection of $K \geq d$ vectors $f_i \in \mathbb{R}^d$ for which there exist two positive frame constants A, B such that for all $v \in \mathbb{R}^d$ we have

$$A\|v\|_2^2 \leq \sum_{i=1}^K |\langle f_i, v\rangle|^2 \leq B\|v\|_2^2. \tag{1}$$

From (1) it follows that F, interpreted as $d \times K$ matrix, has rank d and that its non-zero singular values are in the interval $[\sqrt{A}, \sqrt{B}]$. If B can be chosen equal to A, that is $B = A$, the frame is called **tight**. If all frame elements f_i have unit norm, we call F a unit norm frame. For more details on frames see for instance the introduction by Christensen (2003).
Finally we introduce the Landau symbols O, o to characterise the growth of a function. We write

$$f(t) = O(g(t)) \quad \text{if} \quad \lim_{t\to 0/\infty} f(t)/g(t) = C < \infty$$
$$\text{and} \quad f(t) = o(g(t)) \quad \text{if} \quad \lim_{t\to 0/\infty} f(\varepsilon)/g(\varepsilon) = 0.$$

.

3. A Response Maximization Criterion

One of the origins of dictionary learning can be found in the field of vector quantization, where the aim is to find a codebook (dictionary) such that the codewords (atoms) closely represent the data, that is

$$\min_{\Phi,X} \|Y - \Phi X\|_F^2 \quad \text{s.t.} \quad x_n \in \{e_i\}_i.$$

Indeed the vector quantization problem can be seen as an extreme case of dictionary learning, where we do not only want all our signals to be approximately 1-sparse but also want the single non-zero coefficient equal to one. On the other hand we allow the atoms (codewords) to have any length. The problem above is usually solved by a K-means algorithm, which alternatively separates the training data into K clusters, each assigned to one codeword, and then updates the codeword to be the mean of the associated train signals. For more detailed information about vector quantization or the K-means algorithm see for instance the book by Gersho and Gray (1992) or the introduction by Aharon et al. (2006). If we relax the condition that each coefficient has to be positive, but in turn ask for the atoms to have unit norm, we are already getting closer to the concept of 1-sparse dictionary learning,

$$\min_{\Phi\in\mathcal{D},X} \|Y - \Phi X\|_F^2 \quad \text{s.t.} \quad x_n \in \{\pm e_i\}_i.$$

This minimization problem can be rewritten as

$$\begin{aligned}\min_{\Phi\in\mathcal{D}} \sum_n \min_{i,\sigma_i=\pm 1} \|y_n - \sigma_i\phi_i\|_2^2 &= \min_{\Phi\in\mathcal{D}} \sum_n \min_{i,\sigma_i=\pm 1} \|y_n\|_2^2 - 2\sigma_i\langle y_n,\phi_i\rangle + \|\phi_i\|_2^2 \\ &= \|Y\|_F^2 + N - 2\max_{\Phi\in\mathcal{D}} \sum_n \max_i |\langle y_n,\phi_i\rangle|,\end{aligned}$$

and is therefore equivalent to the maximization problem

$$\max_{\Phi\in\mathcal{D}} \sum_n \max_i |\langle y_n,\phi_i\rangle|. \tag{2}$$

A local maximum of (2) can be found with a signed K-means algorithm, which assigns each training signal to the atom of the current dictionary giving the largest response in absolute value and then updating the atom as normalized signed mean of the associated training signals, see Section 6 for more details. The question now is how do we go from these 1-sparse dictionary learning formulations to S-sparse formulations with $S > 1$. The most common generalization, which provides the starting point for the MOD and the K-SVD algorithm, is to give up all constraints on the coefficients except for S-sparsity and to minimize,

$$(P_{P2}) \qquad \min_{\Phi\in\mathcal{D},X} \|Y - \Phi X\|_F^2 \quad \text{s.t.} \quad \|x_n\|_0 \leq S. \tag{3}$$

However, rewriting the problem we see that this formulation does not reduce to the same maximization problem in case $S = 1$. Then the best one term approximation in the given

dictionary is simply the largest projection onto one atom and we have

$$\begin{aligned}\min_{\Phi\in\mathcal{D}}\sum_n \min_{i,x_i}\|y_n - x_i\phi_i\|_2^2 &= \min_{\Phi\in\mathcal{D}}\sum_n \min_i \|y_n - \langle\phi_i, y_n\rangle\phi_i\|_2^2\\ &= \|Y\|_F^2 - \max_{\Phi\in\mathcal{D}}\sum_n \max_i |\langle y_n, \phi_i\rangle|^2,\end{aligned}$$

leading instead to the maximization problem

$$\max_{\Phi\in\mathcal{D}}\sum_n \max_i |\langle y_n, \phi_i\rangle|^2 \qquad \text{vs.} \qquad \max_{\Phi\in\mathcal{D}}\sum_n \max_i |\langle y_n, \phi_i\rangle|.$$

A local maximum can now be found using the same partitioning strategy as before but updating the atoms as largest singular vector rather than signed mean of the associated training signals, requiring K SVDs as opposed to K means. While the minimization problem (3) is definitely the most effective generalization for dictionary learning when the goal is compression, it brings with it some complications when used as analysis tool. Indeed it has been shown (Schnass, 2014) that for $S = 1$ the K-SVD criterion (3) can only identify the underlying dictionary from sparse random mixtures to arbitrary precision (given enough training samples) if this dictionary is tight and it is conjectured that the same holds for $S \geq 1$. Roughly simplified the reason for this is that for random sparse signals $\Phi_I x_I$ and an ε-perturbation Ψ the average of the largest squared response behaves like

$$\frac{1}{2}\left(1 - \frac{\varepsilon^2}{2} + c(\Psi)\right)^2 + \frac{1}{2}\left(1 - \frac{\varepsilon^2}{2} - c(\Psi)\right)^2 = 1 - \varepsilon^2 + \frac{\varepsilon^4}{4} + c(\Psi)^2.$$

If Φ is tight the term $c(\Psi)$ is constant over all dictionaries and therefore there is a local maximum at Φ. From the above we also see that the average of the largest *absolute* response should behave like

$$\frac{1}{2}\left|1 - \frac{\varepsilon^2}{2} + c(\Psi)\right| + \frac{1}{2}\left|1 - \frac{\varepsilon^2}{2} - c(\Psi)\right| = 1 - \frac{\varepsilon^2}{2},$$

meaning that we should have a maximum at Φ also if it is non-tight. This suggests as alternative way to generalize the K-means optimization principle for dictionary identification to simply maximize the absolute norm of the S-largest responses,

$$(P_{R1}) \qquad \max_{\Psi\in\mathcal{D}}\sum_n \max_{|I|=S}\|\Psi_I^\star y_n\|_1. \tag{4}$$

Other than for the K-SVD criterion it is not obvious that there should be a local optimum of (4) at Φ even if all signals y_n are perfectly S-sparse in Φ. Therefore it is quite intriguing that we will not only be able to prove local identifiability of any generating dictionary via (4) from randomly sparse signals, but that these identifiability properties are stable under coherence and noise. However, before we get to the main result in Theorem 5 on page 1221, we first have to lay the foundation, by providing suitable random signal models and by studying the asymptotic identifiability properties of the new principle.

4. Asymptotic Results

We can get to the asymptotic version of the S-response maximization principle in (4) simply by replacing the sum over the training signals with the expectation, leading to

$$\max_{\Psi \in \mathcal{D}} \mathbb{E}_y \left(\max_{|I|=S} \|\Psi_I^\star y\|_1 \right). \tag{5}$$

Next we need a random sparse coefficient model to generate our signals y. We make the following definition, see also Schnass (2014).

Definition 1 *A probability distribution (measure) ν on the unit sphere $S^{K-1} \subset \mathbb{R}^K$ is called symmetric if for all measurable sets $\mathcal{X} \subseteq S^{K-1}$, for all sign sequences $\sigma \in \{-1,1\}^K$ and all permutations p we have*

$$\begin{aligned} \nu(\sigma\mathcal{X}) = \nu(\mathcal{X}), \quad \textit{where} \quad \sigma\mathcal{X} := \{(\sigma_1 x_1, \dots, \sigma_K x_K) : x \in \mathcal{X}\}, \quad \textit{and} \\ \nu(p(\mathcal{X})) = \nu(\mathcal{X}), \quad \textit{where} \quad p(\mathcal{X}) := \{(x_{p(1)}, \dots, x_{p(K)}) : x \in \mathcal{X}\}. \end{aligned}$$

Setting $y = \Phi x$ where x is drawn from a symmetric probability measure ν on the unit sphere has the advantage that for dictionaries which are orthonormal bases the resulting signals have unit norm and for general dictionaries the signals have unit square norm in expectation, that is $\mathbb{E}(\|y\|_2^2) = 1$. This reflects the situation in practical application, where we would normalize the signals in order to equally weight their importance.
One example of such a probability measure can be constructed from a non-negative, non-increasing sequence $c \in \mathbb{R}^K$ with $\|c\|_2 = 1$, which we permute uniformly at random and provide with random $\pm$ signs. To be precise for a permutation $p : \{1, ..., K\} \to \{1, ..., K\}$ and a sign sequence σ, $\sigma_i = \pm 1$, we define the sequence $c_{p,\sigma}$ component-wise as $c_{p,\sigma}(i) := \sigma_i c_{p(i)}$, and set $\nu(x) = (2^K K!)^{-1}$ if there exist p, σ such that $x = c_{p,\sigma}$ and $\nu(x) = 0$ otherwise. While being very simple this measure exhibits all the necessary structure and indeed in our proofs we will reduce the general case of a symmetric measure to this simple case.
So far we have not incorporated any sparse structure in our coefficient distribution. To motivate the sparsity requirements on our coefficients we will recycle the simple negative example of a sparse coefficient distribution for which the original generating dictionary is not at a local maximum of (5) with $S = 1$ (Schnass, 2014).

Example 1 *Let U be an orthonormal basis and let the signals be constructed as $y = \Phi x$. If x is randomly 2-sparse with 'flat' coefficients, that is, drawn from the simple symmetric probability measure with base sequence c, where $c_1 = c_2 = 1/\sqrt{2}$, $c_i = 0$ for $i \geq 3$, then U is not a local maximum of (5) with $S = 1$.*
Indeed, since the signals are all 2-sparse, the maximal inner product with all atoms in U is the same as the maximal inner product with only $d-1$ atoms. This degree of freedom we can use to construct an ascent direction. Choose $U_\varepsilon = (u_1, \dots, u_{d-1}, (u_d + \varepsilon u_1)/\sqrt{1+\varepsilon^2})$,

then we have

$$\begin{aligned}
\mathbb{E}_y\left(\|U_\varepsilon^\star y\|_\infty\right) &= \mathbb{E}_x\left(\left\|\left(x_1,\ldots,x_{d-1},\frac{x_d+\varepsilon x_1}{\sqrt{1+\varepsilon^2}}\right)\right\|_\infty\right)\\
&= \mathbb{E}_x \max\left\{\frac{1}{\sqrt{2}},\left|\frac{x_d+\varepsilon x_1}{\sqrt{1+\varepsilon^2}}\right|\right\}\\
&= \frac{1}{\sqrt{2}}\left(1-\frac{1}{d(d-1)}+\frac{1}{d(d-1)}\frac{1+\varepsilon}{\sqrt{1+\varepsilon^2}}\right)\\
&\geq \frac{1}{\sqrt{2}}\left(1+\frac{1}{d(d-1)}\frac{\varepsilon-\varepsilon^2}{1+\varepsilon^2}\right) > \frac{1}{\sqrt{2}} = \mathbb{E}_y\left(\|U^\star y\|_\infty\right).
\end{aligned}$$

From the above example we see that, in order to have a local maximum of (5) with $S=1$ at the original dictionary, we need our signals to be truly 1-sparse, that is, we need to have a decay between the first and the second largest coefficient. In the following sections we will study how large this decay should be to have a local maximum exactly at or near to the generating dictionary for more general dictionaries and sparsity levels.

4.1 Exact Recovery

To warm up we first provide an asymptotic exact dictionary identification result for (5) for incoherent dictionaries in the noiseless setting.

Theorem 2 *Let Φ be a unit norm frame with frame constants $A \leq B$ and coherence μ. Let the coefficients x be drawn from a symmetric probability distribution ν on the unit sphere $S^{K-1} \subset \mathbb{R}^K$ and assume that the signals are generated as $y = \Phi x$. If there exists $\beta > 0$ such that for $c_1(x) \geq c_2(x) \geq \ldots \geq c_K(x) \geq 0$ the non-increasing rearrangement of the absolute values of the components of x we have $c_S(x) - c_{S+1}(x) - 2\mu\|x\|_1 \geq \beta$ almost surely, that is*

$$\nu\left(c_S(x) - c_{S+1}(x) - 2\mu\|x\|_1 \geq \beta\right) = 1, \tag{6}$$

then there is a local maximum of (5) at Φ.
Moreover for $\Psi \neq \Phi$ we have $\mathbb{E}_y\left(\max_{|I|=S}\|\Psi_I^\star y\|_1\right) < \mathbb{E}_y\left(\max_{|I|=S}\|\Phi_I^\star y\|_1\right)$ as soon as

$$\varepsilon < \frac{\beta}{1+3\sqrt{\log\left(\frac{25K^2S\sqrt{B}}{\beta(\bar{c}_1+\ldots+\bar{c}_S)}\right)}}, \tag{7}$$

where $\bar{c}_i := \mathbb{E}_x(c_i(x))$.

Proof idea We briefly sketch the main ideas of the proof, which are the same as for the corresponding theorem for the K-SVD principle (Schnass, 2014). For self-containedness of the paper the full proof is included in Appendix A.1.
Assume that we have the case of a simple probability measure based on one sequence c, that is $x = c_{p,\sigma}$. For any fixed permutation p the condition in (6) ensures that for all sign sequences σ, and consequently all signals, the maximal S responses of the original dictionary Φ are attained at $I_p = p^{-1}(\{1\ldots S\})$ and that there is a gap of size β to the remaining responses.

For an ε-perturbation of the generating dictionary we have $\psi_i \approx (1-\varepsilon_i^2/2)\phi_i + \varepsilon_i z_i$ for some unit vectors z_i with $\langle z_i, \phi_i \rangle = 0$ and $\varepsilon_i \leq \varepsilon$. Now for most sign sequences the contribution of $\varepsilon_i z_i$ to the response $\langle \psi_i, \Phi c_{p,\sigma} \rangle$ will be smaller than $\beta/2$ so the maximal S responses will still be attained at I_p. Comparing the loss of the perturbed dictionary over the typical sign sequences of all permutations, which scales as $\frac{(c_1+\ldots+c_S)}{2K} \sum \varepsilon_i^2$, to the maximal gain $S\varepsilon\sqrt{B}$ over the approximately $2\sum_i \exp\left(-\beta^2/\varepsilon_i^2\right)$ atypical sign sequences shows that there is a maximum at the original dictionary. The general result follows from an integration argument. ■

As already mentioned, while for the K-SVD criterion (3) there is always an optimum at the generating dictionary if all training signals are S-sparse, this it is not obvious for the response principle. Indeed, in the special case where all the training signals are exactly S-sparse, $c_{S+1}(x) = 0$ almost surely, we get an additional condition to ensure asymptotic recoverability,

$$c_S(x) - 2\mu \sum_{s=1}^{S} c_s(x) \geq \beta > 0, \qquad \text{almost surely.}$$

To get a better feeling for this constraint we bound the sum over the S largest responses by S times the largest response, $\sum_{s=1}^{S} c_s(x) \leq S c_1(x)$ and arrive at the condition

$$\frac{c_S(x)}{c_1(x)} \gtrsim 2S\mu, \tag{8}$$

which is the classical condition under which simple thresholding will find the support of an exactly S-sparse signal (Schnass and Vandergheynst, 2008).

4.2 Stability under Coherence and Noise

While giving a first insight into the identification properties of the response principle, Theorem 2 suffers from two main limitations.
First, the required condition on the coherence of the dictionary with respect to the decay of the coefficients, $c_S(x) - c_{S+1}(x) - 2\mu\|c(x)\|_1 > 0$, is unfortunately quite strict. In the most favourable case of exactly S sparse signals with equally sized coefficients, $c_1(x) = c_S(x) = 1/\sqrt{S}$, we see from (8) that we can only identify dictionaries from very sparse signals, where $S =\lesssim \mu^{-1}$. In case of very incoherent dictionaries with $\mu = O(1/\sqrt{d})$ this means that $S \lesssim \sqrt{d}$. However, for most sign sequences σ we have

$$|\langle \phi_i, \Phi c_{p,\sigma} \rangle| = \left| \sigma_i c_{p(i)} + \sum_{j \neq i} \sigma_j c_{p(j)} \langle \phi_i, \phi_j \rangle \right| \approx c_{p(i)} \pm \left(\sum_{j \neq i} c_{p(j)}^2 |\langle \phi_i, \phi_j \rangle|^2 \right)^{1/2} \approx c_{p(i)} \pm \mu,$$

which indicates that a condition of the form $\mu \lesssim c_S - c_{S+1}$ may be strong enough to guarantee (approximate) recoverability of the dictionary. Assuming again the most favourable case of equally sized coefficients, we could therefore identify dictionaries from signals with sparsity levels of the order $S \lesssim \mu^{-2}$, which, in case of incoherent dictionaries, means of the order of the ambient dimension $S \lesssim d$.

The second limitation of Theorem 2 is that, even if it allows for not exact S-sparseness of the signals, it does not take into account noise. Our next goal is therefore to extend the exact identification result in Theorem 2 to a stable identification result for less sparse (larger S) and noisy signals. For this task we first need to amend our signal model to incorporate noise. We would like to consider unbounded white noise, but also keep the property that in expectation the signals have unit square norm. Further for the next section, where we want to transform our asymptotic identification results to results for finite sample sizes, it will be convenient if our signals are bounded. These considerations lead to the following model:

$$y = \frac{\Phi x + r}{\sqrt{1 + \|r\|_2^2}}, \tag{9}$$

where $r = (r(1) \dots r(d))$ is a centred random subgaussian vector with parameter ρ. That is, the entries $r(i)$ are independent and satisfy $\mathbb{E}(e^{t \cdot r(i)}) \leq e^{t^2 \rho^2 / 2}$.
Employing this noisy signal model and formalizing the ideas about the typical gap size between responses of the generating dictionary inside and outside the true support, leads to the following theorem.

Theorem 3 *Let Φ be a unit norm frame with frame constants $A \leq B$ and coherence μ. Let the coefficients x be drawn from a symmetric probability distribution ν on the unit sphere $S^{K-1} \subset \mathbb{R}^K$. Further let $r = (r(1) \dots r(d))$ be a centred random subgaussian noise-vector with parameter ρ and assume that the signals are generated according to the noisy signal model in (9). If there exists $\beta > 0$ such that for $c_1(x) \geq c_2(x) \geq \dots \geq c_K(x) \geq 0$ the non-increasing rearrangement of the absolute values of the components of x we have $c_S(x) - c_{S+1}(x) \geq \beta$ almost surely and*

$$\max\{\mu, \rho\} \leq \frac{\beta}{\sqrt{72(\log a + \log\log a)}} \quad \text{for} \quad a = \frac{112 K^2 S(\sqrt{B} + 1)}{C_r \beta(\bar{c}_1 + \dots + \bar{c}_S)}, \tag{10}$$

where $C_r = \mathbb{E}_r\left((1 + \|r\|_2^2)^{-1/2}\right)$ and $\bar{c}_i := \mathbb{E}_x(c_i(x))$, then there is a local maximum of (5) at $\tilde{\Psi}$ satisfying

$$d(\tilde{\Psi}, \Phi) \leq \frac{12 S K^2 \sqrt{B}}{C_r(\bar{c}_1 + \dots + \bar{c}_S)} \exp\left(\frac{-\beta^2}{72 \max\{\mu^2, \rho^2\}}\right). \tag{11}$$

Proof idea As outlined at the beginning of the section the main ingredient we have to add to the proof idea of Theorem 2 is a probabilistic argument to substitute the condition guaranteeing that the S largest responses of the generating dictionary are I_p. Due to concentration of measure we get that for most sign sequences, and therefore most signals, the maximum is still attained at I_p. Moreover the gap to the remaining responses is actually large enough to accommodate relatively high levels of noise and/or perturbations.
The detailed proof can be found in Appendix A.2. ■

Let us make some observations about the last result.
First, we want to point out that for sub-Gaussian noise with parameter ρ, the quantity $C_r = \mathbb{E}_r\left((1 + \|r\|_2^2)^{-1/2}\right)$ in the statement above is well behaved. If for example the $r(i)$ are iid Bernoulli-variables, that is $P(r(i) = \pm\rho) = \frac{1}{2}$, we have $C_r = (1 + d\rho^2)^{-1/2}$. In general

we have the following estimate due for instance to Theorem 1 by Hsu et al. (2012). Since we have

$$\mathbb{P}\left(\|x\|_2^2 \geq \rho^2(d+2\sqrt{dt}+2t\right) \leq e^{-t},$$

setting $t=d$, we get $\mathbb{P}\left(\|x\|_2^2 \geq 5d\rho^2\right) \leq e^{-d}$, which leads to

$$\mathbb{E}_r\left(\frac{1}{\sqrt{1+\|r\|_2^2}}\right) \geq \frac{(1-e^{-d})}{\sqrt{1+5d\rho^2}}.$$

Also to illustrate the result we again specialize it to the most favourable case of exactly S-sparse signals with balanced coefficients, that is $c_S(x) = S^{-1/2}$. Assuming white Gaussian noise with variance ρ_G^2 we see that identification is possible even for expected signal to noise ratios of the order $O(\frac{S}{d})$, that is

$$\frac{\mathbb{E}(\|\Phi x\|_2^2)}{\mathbb{E}(\|r\|_2^2)} \gtrsim \frac{S}{d}.$$

Similarly, by specializing Theorem 3 to the case of exactly S-sparse and noiseless signals we get - to the best of our knowledge - the first result establishing that locally it is possible to stably identify dictionaries from signals with sparsity levels beyond the spark of the generating dictionary. Indeed, even if some of the S-sparse signals could have representations in Φ that require less than S atoms, there will still be a local maximum of the asymptotic criterion close to the original dictionary as long as the smallest coefficient of each signal is of the order $O(\mu)$, which in the most favourable case means that we can have $S \lesssim \mu^{-2}$ or $S \lesssim d$. The quality of this result is on a par with the best results for finding sparse approximations in a *given* dictionary, which say that on average Basis Pursuit or thresholding can find the correct sparse support even for signals with sparsity levels of the order of the ambient dimension (Tropp, 2008; Schnass and Vandergheynst, 2007).
Next note that with the available tools it would be possible to consider also a signal model where a small fraction of the coefficients violates the decay condition $c_S(x) - c_{S+1}(x) \geq \beta$ and still have stability. However, we leave explorations in that direction to the interested reader and instead turn to the study of the criterion for a finite number of training samples.

5. Finite Sample Size Results

In this section we will transform the two asymptotic results from the last section into results for finite sample sizes, that is, we will study when Φ is close to a local maximum of

$$\max_{\Psi\in\mathcal{D}} \frac{1}{N}\sum_{n=1}^{N} \max_{|I|=S} \|\Psi_I^\star y_n\|_1, \tag{12}$$

assuming that the y_n are following either the noise-free or the noisy signal model. For convenience we will do the analysis for the normalized version (12) of the S-response criterion (4).

Theorem 4 *Let Φ be a unit norm frame with frame constants $A \leq B$ and coherence μ. Let the coefficients x_n be drawn from a symmetric probability distribution ν on the unit sphere*

$S^{K-1} \subset \mathbb{R}^K$ and assume that the signals are generated as $y_n = \Phi x_n$. If there exists $\beta > 0$ such that for $c_1(x_n) \geq c_2(x_n) \geq \ldots \geq c_K(x_n) \geq 0$ the non-increasing rearrangement of the absolute values of the components of x_n we have $c_S(x_n) - c_{S+1}(x_n) - 2\mu\|x_n\|_1 \geq \beta$ almost surely and the target precision $\tilde{\varepsilon}$ satisfies

$$\tilde{\varepsilon} \leq \frac{\beta}{1 + 3\sqrt{\log\left(\frac{50K^2S\sqrt{B}}{\beta(\bar{c}_1+\ldots+\bar{c}_S)}\right)}},$$

where $\bar{c}_i := \mathbb{E}_{x_n}(c_i(x_n))$, then except with probability,

$$2\exp\left(-\frac{N\tilde{\varepsilon}^2(\bar{c}_1+\ldots+\bar{c}_S)^2}{129S^2K^2B} + Kd\log\left(\frac{25SK\sqrt{B}}{\tilde{\varepsilon}(\bar{c}_1+\ldots+\bar{c}_S)}\right)\right),$$

there is a local maximum of (12) respectively (4) at $\tilde{\Psi}$ satisfying

$$d(\tilde{\Psi}, \Phi) \leq \tilde{\varepsilon} + \frac{\tilde{\varepsilon}^2}{4K}.$$

Theorem 5 *Let Φ be a unit norm frame with frame constants $A \leq B$ and coherence μ. Let the coefficients x_n be drawn from a symmetric probability distribution ν on the unit sphere $S^{K-1} \subset \mathbb{R}^K$. Further let $r_n = (r_n(1)\ldots r_n(d))$ be i.i.d. centred random subgaussian noise-vectors with parameter ρ and assume that the signals are generated according to the noisy signal model in (9). If there exists $\beta > 0$ such that for $c_1(x_n) \geq c_2(x_n) \geq \ldots \geq c_K(x_n) \geq 0$ the non-increasing rearrangement of the absolute values of the components of x we have $c_S(x_n) - c_{S+1}(x_n) \geq \beta$ almost surely and if the target precision $\tilde{\varepsilon}$, the noise parameter ρ and the coherence μ satisfy*

$$\tilde{\varepsilon} \leq \frac{\beta}{\frac{9}{4} + 9\sqrt{\log a}} \quad \text{and} \tag{13}$$

$$\max\{\mu, \rho\} \leq \frac{\beta}{\sqrt{72(\log a + \log\log a)}} \quad \text{for} \quad a = \frac{150K^2S(\sqrt{B}+1)}{C_r\beta(\bar{c}_1+\ldots+\bar{c}_S)},$$

where $C_r = \mathbb{E}_{r_n}\left((1+\|r_n\|_2^2)^{-1/2}\right)$ and $\bar{c}_i := \mathbb{E}_{x_n}(c_i(x_n))$, then except with probability

$$2\exp\left(-\frac{N\tilde{\varepsilon}_{\mu,\rho}^2(\bar{c}_1+\ldots+\bar{c}_S)^2}{513C_r^2S^2K^2\left(\sqrt{B}+1\right)^2} + Kd\log\left(\frac{49SK\left(\sqrt{B}+1\right)}{\varepsilon_{\mu,\rho}(\bar{c}_1+\ldots+\bar{c}_S)}\right)\right),$$

$$\text{where} \qquad \tilde{\varepsilon}_{\mu,\rho} = \max\left\{\tilde{\varepsilon}, \frac{16SK^2\left(\sqrt{B}+1\right)}{C_r(\bar{c}_1+\ldots+\bar{c}_S)}\exp\left(-\frac{\beta^2}{72\max\{\mu^2,\rho^2\}}\right)\right\},$$

there is a local maximum of (12) respectively (4) at $\tilde{\Psi}$, satisfying

$$d(\tilde{\Psi}, \Phi) \leq \tilde{\varepsilon}_{\mu,\rho} + \frac{\tilde{\varepsilon}_{\mu,\rho}^2}{16K}.$$

Proof idea The proofs, which can be found in Appendix A.3, are based on three ingredients, a Lipschitz property for the mapping $\Psi \to \frac{1}{N}\sum_{n=1}^{N} \max_{|I|=S} \|\Psi_I^\star y_n\|_1$ for the respective signal model, the concentration of the sum around its expectation for a δ-net covering the space of all admissible dictionaries close to Φ and a triangle inequality argument to show that the finite sample response differences are close to the expected response differences and therefore larger than 0 for all $\varepsilon \gtrsim \varepsilon_{\mu,\rho}$. ■

To see better how the sample complexity behaves, we simplify the two theorems to the special case of noiseless exactly S-sparse signals with balanced coefficients for various orders of magnitude of S.

If we have $S = O(1)$, Theorem 4 implies that in order to have a maximum within radius $\tilde{\varepsilon}$ to the original dictionary Φ with probability e^{-Kd} we need $N = O\big(K^3 d\tilde{\varepsilon}^{-2}\big)$ samples. Conversely given N training signals we can expect the distance between generating dictionary and closest local maximum to be of the order $O\big(K^2 N^{-1/2}\big)$.

If we assume a very incoherent dictionary where $\mu = O(d^{-1/2})$ and thus let the sparsity level be of the order $O(\sqrt{d})$ the sample complexity rises to $N = O\big(K^3 d^{3/2}\tilde{\varepsilon}^{-2}\big)$. Taking into account that by (13) the target precision $\tilde{\varepsilon}$ needs to be of order $O\left(S^{-1/2}\right) = O\left(d^{-1/4}\right)$ this means that we need at least $N = O\big(K^3 d^2\big)$ training signals and once this initial level is reached, the error goes to zero at rate $N^{-1/2}$.

For an even lower sparsity level, $S = O(d)$, again assuming a very incoherent dictionary, the sample complexity for target precision $\tilde{\varepsilon}$ implied now by Theorem 5 rises to $N = O\big(K^3 d^2 \tilde{\varepsilon}^{-2}\big)$. In this regime, however, we cannot reach arbitrarily small errors by choosing N large enough but only approach the asymptotic precision $\tilde{\varepsilon}_\mu = 16K^2\sqrt{SB}\exp\left(-d/72S\right)$.

Following these promising theoretical results, in the next section we will finally see how theory translates into practice.

6. Experiments

After showing that the optimization criterion in (4) is locally suitable for dictionary identification, in this section we present an iterative thresholding and K means type algorithm (ITKM) to actually find the local maxima of (4) and conduct some experiments to illustrate the theoretical results. We recall that given the input signals $Y = (y_1 \dots y_N)$ and a fixed sparsity parameter S we want to solve,

$$\max_{\Psi \in \mathcal{D}} \sum_n \max_{|I|=S} \|\Psi_I^\star y_n\|_1.$$

Using Lagrange multipliers,

$$\frac{\partial}{\partial \psi_k}\left(\sum_n \max_{|I|=S} \|\Psi_I^\star y_n\|_1\right) = \sum_{n:k\in I(\Psi,y_n)} \operatorname{sign}(\langle \psi_k, y_n\rangle) y_n^\star,$$

$$\frac{\partial}{\partial \psi_k}\left(\|\psi_k\|_2^2\right) = 2\psi_k^\star,$$

where $I(\Psi, y_n) := \operatorname{argmax}_{|I|=S} \|\Phi_I^\star y_n\|_1$, we arrive at the following update rule,

$$\psi_k^{new} = \lambda_k \cdot \sum_{n:k\in I(\Psi^{old},y_n)} \operatorname{sign}(\langle \psi_k^{old}, y_n\rangle) y_n, \tag{14}$$

where λ_k is a scaling parameter ensuring that $\|\psi_k^{new}\|_2 = 1$.
In practice, when we do not have an oracle giving us the generating dictionary as initialization, we also need to safeguard against bad initializations resulting in a zero-update $\psi_k^{new} = 0$. For example we can choose the zero-updated atom uniformly at random from the unit sphere or from the input signals.
Note that finding the sets $I(\Psi^{old}, y_n)$ corresponds to N thresholding operations while updating according to (14) corresponds to K signed means. Altogether this means that each iteration of ITKM has computational complexity determined by the matrix multiplication $\Psi^\star Y$, meaning $O(dKN)$. This is light in comparison to K-SVD, which even when using thresholding instead of OMP as sparse approximation procedure still requires the calculation of the maximal singular vector of K on average $d \times \frac{N}{K}$ matrices. It is also more computationally efficient than the algorithm for local dictionary refinement, proposed by Arora et al. (2014), which is also based on averaging. Furthermore it is straightforward to derive online or parallelized versions of ITKM. In an online version for each newly arriving signal y_n we calculate $I(\Psi^{old}, y_n)$ using thresholding and update $\psi_k^{new} = \psi_k^{new} + \operatorname{sign}(\langle \psi_k^{old}, y_n\rangle) y_n$ for $k \in I(\Psi^{old}, y_n)$. After N signals have been processed we renormalize the atoms ψ_k^{new} to have unit norm and set $\Psi^{old} = \Psi^{new}$. Similarly, to parallelize we divide the training samples into m sets of size $\frac{N}{m}$. Then on each node m we learn a dictionary Ψ_m^{new} according to (14) with $\lambda_k = 1$. We then calculate the sum of these dictionaries $\Psi_0^{new} = \sum_m \Psi_m^{new}$ and renormalize the atoms in Ψ_0^{new} to have unit norm.
Armed with this very simple algorithm we will now conduct four experiments to illustrate our theoretical findings[1].

6.1 ITKM vs. K-SVD

In our first experiment we compare the local recovery error of ITKM and K-SVD for 3-dimensional bases with increasing condition numbers.
The bases are perturbations of the canonical basis $\Phi = (e_1, e_2, e_3)$ with the vector $v = (1, 1, 1)$. That is, $\Phi^t = (e_1^t, e_2^t, e_3^t)$, where $e_i^t = (e_i + tv)/\|(e_i + tv)\|_2$ and t varying from 0 to 0.5 in steps of 0.1, which corresponds to condition numbers $\kappa(\Phi^t)$ varying from 1 to 2.5. We generate $N = 4096$ approximately 1-sparse noiseless signals from the signal model described in Table 1 with $S = 1$, $T = 2$, $\rho = 0$ and $b = 0.1/0.2$ and run both ITKM and K-SVD with 1000 iterations, sparsity parameter $S = 1$ and the true dictionary (basis) as initialization. Figure 1(a) shows the recovery error $d(\Phi^t, \tilde{\Psi})$ between the original dictionary and the output of the respective algorithm averaged over 10 runs.
As predicted by the theoretical results on the corresponding underlying minimization principles, the recovery errors of ITKM and K-SVD are roughly the same for Φ^0, which is an orthogonal basis and therefore tight. However, while for ITKM the recovery error stays

1. A Matlab penknife (mini-toolbox) for playing around with ITKM and reproducing the experiments can be found at `http://homepage.uibk.ac.at/~c7021041/ITKM.zip`.

Signal Model
Given the generating dictionary Φ our signal model further depends on four coefficient parameters,

S	-	the effective sparsity or number of comparatively large coefficients,
b	-	deciding the decay factor of these sparse coefficients,
T	-	the total number of non-zero coefficients ($T \geq S$) and
ρ	-	the noise level.

Given these parameters we choose a decay factor c_b uniformly at random in the interval $[1-b, 1]$. We set $c_i = c_b^i/\sqrt{S}$ for $1 \leq i \leq S$ and $c_i = 0$ for $T < i \leq K$. If $T = S$ we renormalize the sequence to have unit norm, while if $T > S$ we choose the vector $(c_{S+1}, \ldots, c_T)$ uniformly at random on the sphere of radius R, where R is chosen such that the resulting sequence c has unit norm. We then choose a permutation p and a sign sequence σ uniformly at random and set $y = \Phi c_{p,\sigma}$, respectively $y = (\Phi c_{p,\sigma} + r)/\sqrt{1 + \|r\|_2}$ where r is a Gaussian noise-vector with variance ρ^2 if $\rho > 0$.

Table 1: Signal Model

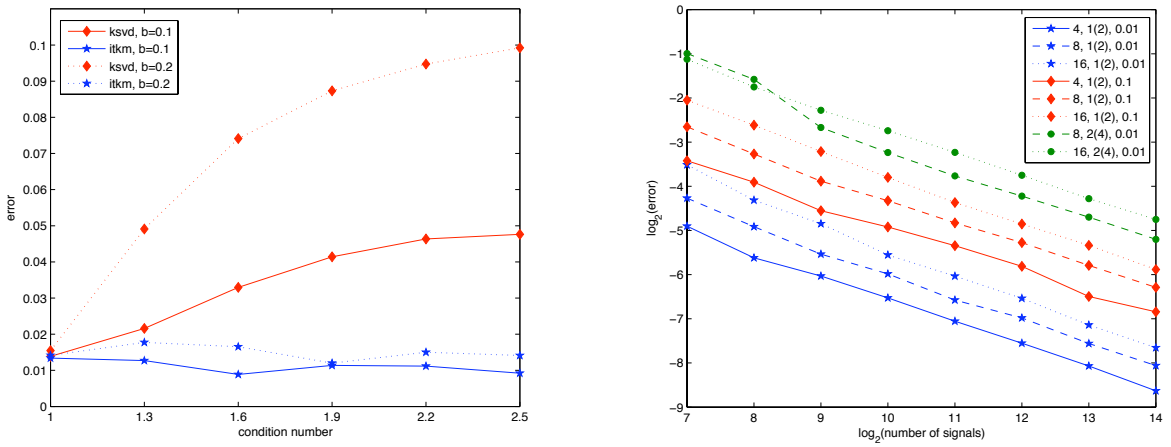
constantly low over all condition numbers, for K-SVD it increases with increasing condition number or non-tightness.

6.2 Recovery Error and Sample Size

The next experiment is designed to show how fast the maximizer $\tilde{\Psi}$ near the original dictionary Φ converges to Φ with increasing sample size N.
The generating dictionaries consist of the canonical basis in $\mathbb{R}^d$ for $d = 4, 8, 16$ and the first $d/2$ elements of the Hadamard basis and as such are not tight. For every set of parameters $d, S(T), b$ we generate N noiseless signals with N varying from $2^7 = 128$ to $2^{14} = 16384$ and run ITKM with 1000 iterations, sparsity parameter S equal to the coefficient parameter S and the true dictionary as initialization. Figure 1(b) shows the recovery error $d(\Phi, \tilde{\Psi})$ between the original dictionary Φ and the output of ITKM $\tilde{\Psi}$ averaged over 10 runs.
As predicted by Theorem 4 the recovery error decays as $N^{-1/2}$. However, the separation of the curves for $d = 4, 8, 16$ and almost exactly sparse signals ($b = 0.01$) by a factor around $\sqrt{2}$ instead of 4, as suggested by the estimate $\tilde{\varepsilon} \approx K^2 N^{-1/2}$, indicates that the cubic dependence of the sampling complexity on the number of atoms K may be too pessimistic and could be lowered.

6.3 Stability of Recovery Error under Coherence and Noise

With the last two experiments we illustrate the stability of the maximization criterion under coherence and noise. As generating dictionaries we use again the canonical basis plus half Hadamard dictionaries described in the last experiment, which have coherence $\mu = d^{-1/2}$.To test the stability under coherence we use a large enough number of noiseless training signals $N = 16384$, such that the distance between the local maximum of the criterion near the generating dictionary, that is the output of ITKM with oracle initialization, and the generating dictionary is mainly determined by the ratio between the gap size β and the coherence.

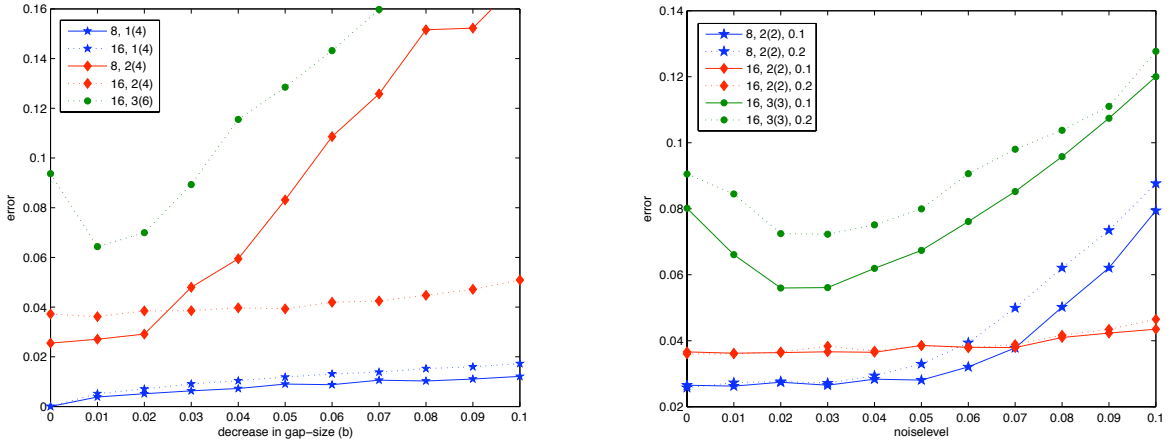


(a) (b)

Figure 1: (a) Local recovery error of K-SVD and ITKM for two different types of decaying coefficients and bases with varying condition numbers in $\mathbb{R}^3$, (b) Decay of recovery error of ITKM with increasing number of training signals

For each set of parameters $d, S(T)$ we create N training signals with decreasing gap sizes β by increasing b from 0 to 0.1 in steps of 0.01 and run ITKM with oracle initialization, parameter S and 1000 iterations. Figure 2(a) shows the recovery error $d(\Phi, \tilde{\Psi})$ between the original dictionary Φ and the output of ITKM $\tilde{\Psi}$ again averaged over 10 trials.
Again the experiments reflect our theoretical results. For $d = 8, 16$ with $S = 1$ or $d = 16$ with $S = 2$ the gap size is large enough that over the whole range of parameters the recovery error stays constantly low at the level defined by the number of samples. Note that this is quite good, since for $b = 0.1$ we are already far beyond the gap size coherence ratios where the stable theoretical results hold. On the other hand for $d = 8$ with $S = 2$ or $d = 16$ with $S = 3$ early on the gap decreases enough to become the error determining factor and so we see an increase in recovery error as b grows.
Conversely to test the stability under noise we use a large enough number of exactly sparse training signals, such that the recovery error will be mainly determined by the noise level. For each set of parameters $d, S(S), b$ we create $N = 16384$ training signals with Gaussian noise of variance (noise level) ρ^2 going from 0 to 0.1 in steps of 0.01 and run ITKM with oracle initialization, parameter S and 1000 iterations. Figure 2(b) shows the recovery error $d(\Phi, \tilde{\Psi})$ between the original dictionary Φ and the output of ITKM $\tilde{\Psi}$, this time averaged over 20 trials.

The curves again correspond to the prediction of the theoretical results, that is the recovery error stays at roughly the same level defined by the number of samples until the noise becomes large enough and then increases. What is maybe interesting to observe in both experiments is the dithering effect for $d = 16$ with $S = 3$, which is due to the special structure of the dictionary. Indeed using almost equally sized, almost exactly sparse coefficients, it is possible to build signals using only the canonical basis that have almost the



(a) (b)

Figure 2: Increase of recovery error with (a) decreasing ratio between coefficient gap and coherence and (b) increasing noise level

same response in only half the Hadamard basis and the other way round. This indicates that slight perturbations of one with the other lead to even better responses and therefore a larger recovery error. After showing that the theoretical results translate into algorithmic practice, we finally turn to a discussion of our results in the context of existing work and point out directions of future research.

7. Discussion

We have introduced a new response maximization principle for dictionary learning and shown that this is locally suitable to identify a generating μ-coherent dictionary from approximately S-sparse training signals to arbitrary precision as long as the sparsity level is of the order $O(\mu^{-1})$. We have also presented - to the best of our knowledge - the first results showing that stable dictionary identification is locally possible not only for signal to noise ratios of the order $O(\sqrt{d})$ but also for sparsity levels of the order $O(\mu^{-2})$.
The derived sample complexity (omitting log factors) of $O(K^3 d\tilde{\varepsilon}^{-2})$, for signals with sparsity levels $S = O(1)$ is roughly the same as for the K-SVD criterion (Schnass, 2014) or the ℓ_1-minimization criterion (Jenatton et al., 2014) but somewhat large compared to recently developed dictionary algorithms that have a sample complexity of $O(K^2)$ (Arora et al., 2014; Agarwal et al., 2014a) or $O(K\varepsilon^{-2})$ (Agarwal et al., 2014b). However, as the sparsity approaches and goes beyond $\mu^{-1} \sim \sqrt{d}$ the derived sample complexity of $O(K^3 d^2 \tilde{\varepsilon}^{-2})$ compares quite favourably to the sample complexity of $O(K^{1/(4\eta)})$ for a sparsity level $d^{1/2-\eta}$ as projected by Arora et al. (2014). Given that also our experimental results suggest that $O(K^3 d\tilde{\varepsilon}^{-2})$ is quite pessimistic, one future direction of research aims to lower the sample complexity. In particular ongoing work suggests that for the ITKM *algorithm* a sample size of order $K \log K$ is enough to guarantee local recovery with high probability.

Another strong point of the results is that the corresponding maximization algorithm ITKM (Iterative Thresholding and K signed Means) is locally successful, as demonstrated in several experiments, and computationally very efficient. The most complex step in each iteration is the matrix multiplication $\Phi^\star Y$ of order $O(dKN)$, which is even lighter than the iterative averaging algorithm described by Arora et al. (2014).
However, the serious drawback is that ITKM is only a local algorithm and that all our results are only local. Also while for the K-SVD criterion and the ℓ_1-minimization criterion there is reason to believe that all local minima might be equivalent, the response maximization principle has a lot of smaller local maxima, which is confirmed by preliminary experiments with random initializations. There ITKM fails but with grace, that is, it outputs local maximizers that have not all, but only most atoms in common with what seems to be the global maximizer near the generating dictionary. This behaviour is in strong contrast to the algorithms presented by Arora et al. (2014); Agarwal et al. (2014b), that have global success guarantees at a computational cost of the order $O(dN^2)$, and leads to several very important research directions.
First we want to confirm that ITKM has a convergence radius of the order $O(1/\sqrt{S})$. This is suggested by the derived radius of the area on which the generating dictionary is the optimal maximizer as well as preliminary experiments. Alternatively, we could investigate how the results for the local iterative algorithms (Arora et al., 2014; Agarwal et al., 2014a) could be extended to larger sparsity levels and convergence radii using our techniques. The associated important question is how to extend the results for the algorithms presented by Arora et al. (2014); Agarwal et al. (2014b) to sparsity levels $O(\mu^{-2})$, if possible at lower cost than $O(dN^2)$. Given the conjectured size of the convergence radius for ITKM it would even be sufficient for the output of the algorithm to arrive at a dictionary within distance $O(1/\sqrt{S})$ to the generating dictionary, since the output could then be used as initialization for ITKM.
A parallel approach for getting global identification results for sparsity levels $O(\mu^{-2})$, that we are currently pursuing, is to analyze a version of ITKM using residual instead of pure signal means, which in preliminary experiments exhibits global convergence properties.
The last research directions we want to point out are concerned with the realism of the signal model. The fact that for an input sparsity S a gap of order $O(\mu^{-2})$ between the S and $S+1$ largest coefficient is sufficient can be interpreted as a relaxed dependence of the algorithm on the sparsity parameter, since a gap of order μ^{-2} can occur quite frequently. To further decrease this sensitivity to the sparsity parameter in the criterion and the algorithm we would therefore like to extend our results to the case where we can only guarantee a gap of order $O(\mu^{-2})$ between the S largest and the $S+T$ largest coefficient for some $T > 1$.
Last but not least we would like to exactly reflect the practical situation, where we would normalize our training signals to equally weight their importance and analyze the unit norm signal model where $y = \Phi x + r/\|\Phi x + r\|_2$.

Acknowledgments

This work was supported by the Austrian Science Fund (FWF) under Grant no. J3335 and improved thanks to the reviewers' comments and suggestions. Thanks also go to the

Computer Vision Laboratory of the University of Sassari, Italy, which provided the surroundings, where almost all of the presented work was done, and to Michael Dymond, who looked over the final manuscript with a native speaker's eye.

Appendix A. Proofs

We now provide the full proofs for all our results.

A.1 Proof of Theorem 2

We first reformulate and prove the theorem for the simple case of a symmetric coefficient distribution based on one sequence and then use an integration argument to extend it to the general case.

Proposition 6 *Let Φ be a unit norm frame with frame constants $A \leq B$ and coherence μ. Let $x \in \mathbb{R}^K$ be a random permutation of a sequence c, where $c_1 \geq c_2 \geq c_3 \ldots \geq c_K \geq 0$ and $\|c\|_2 = 1$, provided with random $\pm$ signs, that is $x = c_{p,\sigma}$ with probability $\mathbb{P}(p,\sigma) = (2^K K!)^{-1}$. Assume that the signals are generated as $y = \Phi x$. If c satisfies $c_S > c_{S+1} + 2\mu\|c\|_1$ then there is a local maximum of (5) at Φ.*
Moreover for $\Psi \neq \Phi$ we have $\mathbb{E}_y\left(\max_{|I|=S} \|\Psi_I^\star y\|_1\right) < \mathbb{E}_y\left(\max_{|I|=S} \|\Phi_I^\star y\|_1\right)$ as soon as

$$d(\Phi,\Psi) \leq \frac{c_S - c_{S+1} - 2\mu\|c\|_1}{1 + 3\sqrt{\log\left(\frac{25K^2S\sqrt{B}}{(c_S - c_{S+1} - 2\mu\|c\|_1)(c_1+\ldots+c_S)}\right)}}. \tag{15}$$

Proof We start by evaluating the objective function at the original dictionary Φ.

$$\mathbb{E}_y\left(\max_{|I|=S} \|\Phi_I^\star y\|_1\right) = \mathbb{E}_p\mathbb{E}_\sigma\left(\max_{|I|=S} \|\Phi_I^\star \Phi c_{p,\sigma}\|_1\right) = \mathbb{E}_p\mathbb{E}_\sigma\left(\max_{|I|=S} \sum_{i\in I} |\langle \phi_i, \Phi c_{p,\sigma}\rangle|\right).$$

To estimate the sum of the (in absolute value) largest S inner products, we first assume that p is fixed. Setting $I_p = p^{-1}(\{1,\ldots S\})$ we have,

$$|\langle \phi_i, \Phi c_{p,\sigma}\rangle| = \left|\sigma_i c_{p(i)} + \sum_{j\neq i} \sigma_j c_{p(j)} \langle \phi_i, \phi_j\rangle\right| \begin{array}{ll} \geq c_S - \mu\|c\|_1 & \forall i \in I_p \\ \leq c_{S+1} + \mu\|c\|_1 & \forall i \notin I_p \end{array}.$$

Together with the condition that $c_S > c_{S+1} + 2\mu\|c\|_1$ these estimates ensure that the S maximal inner products in absolute value are attained at I_p and so we get for the expectation,

$$\begin{aligned}\mathbb{E}_p\mathbb{E}_\sigma\left(\max_{|I|=S} \|\Phi_I^\star \Phi c_{p,\sigma}\|_1\right) &= \mathbb{E}_p\mathbb{E}_\sigma\left(\|\Phi_{I_p}^\star \Phi c_{p,\sigma}\|_1\right) \\ &= \mathbb{E}_p\mathbb{E}_\sigma\left(\sum_{i\in I_p}\left|c_{p(i)} + \sigma_i \sum_{j\neq i} \sigma_j c_{p(j)}\langle \phi_i, \phi_j\rangle\right|\right) = c_1 + \ldots + c_S.\end{aligned}$$

To compute the expectation for a perturbation of the original dictionary we use the following parameterization of all ε-perturbations Ψ of the original dictionary Φ. If $d(\Psi,\Phi) = \varepsilon$ then

$\|\psi_i - \phi_i\|_2 = \varepsilon_i$ with $\max_i \varepsilon_i = \varepsilon$ and we have z_i with $\langle \phi_i, z_i \rangle = 0, \|z_i\|_2 = 1$ and $\alpha_i := 1 - \varepsilon_i^2/2$ and $\omega_i := (\varepsilon_i^2 - \varepsilon_i^4/4)^{\frac{1}{2}}$, such that

$$\psi_i = \alpha_i \phi_i + \omega_i z_i.$$

Expanding the expectation as before we get,

$$\mathbb{E}_y \left(\max_{|I|=S} \|\Psi_I^\star y\|_1 \right) = \mathbb{E}_p \mathbb{E}_\sigma \left(\max_{|I|=S} \|\Psi_I^\star \Phi c_{p,\sigma}\|_1 \right) = \mathbb{E}_p \mathbb{E}_\sigma \left(\max_{|I|=S} \sum_{i \in I} |\langle \psi_i, \Phi c_{p,\sigma} \rangle| \right). \quad (16)$$

The tried and tested strategy applied now is showing that for small perturbations and most sign patterns σ the maximal inner products are still attained by $i \in I_p$. We have

$$\begin{aligned} \forall i \in I_p: \quad & |\langle \psi_i, \Phi c_{p,\sigma} \rangle| \geq \alpha_i (c_S - \mu \|c\|_1) - \omega_i |\langle z_i, \Phi c_{p,\sigma} \rangle| \\ \forall i \notin I_p: \quad & |\langle \psi_i, \Phi c_{p,\sigma} \rangle| \leq \alpha_i (c_{S+1} + \mu \|c\|_1) + \omega_i |\langle z_i, \Phi c_{p,\sigma} \rangle|. \end{aligned}$$

Using Hoeffding's inequality we can estimate the typical sizes of the terms $|\langle z_i, \Phi c_{p,\sigma} \rangle|$,

$$\begin{aligned} \mathbb{P}(|\langle z_i, \Phi c_{p,\sigma} \rangle| \geq t) &= \mathbb{P}(|\sum_{j \neq i} \sigma_j c_{p(j)} \langle z_i, \phi_j \rangle| > t) \\ &\leq 2 \exp\left(- \frac{t^2}{2 \sum_{j \neq i} c_{p(j)}^2 \langle z_i, \phi_j \rangle^2} \right) \leq 2 \exp\left(- \frac{t^2}{2} \right). \end{aligned}$$

In case $\omega_i \neq 0$ or equivalently $\varepsilon_i \neq 0$, we set $t = s/\omega_i$ to arrive at

$$\mathbb{P}(\omega_i |\langle z_i, \Phi c_{p,\sigma} \rangle| \geq s) \leq 2 \exp\left(- \frac{s^2}{2\omega_i^2} \right) \leq 2 \exp\left(- \frac{s^2}{2\varepsilon_i^2} \right),$$

where we have used that $\omega_i^2 = \varepsilon_i^2 - \varepsilon_i^4/4 \leq \varepsilon_i^2$, while in case $\varepsilon_i = 0$ we trivially have that $\mathbb{P}(\omega_i |\langle z_i, \Phi c_{p,\sigma} \rangle| \geq s) = 0$. Summarizing these findings we see that except with probability

$$\eta := 2 \sum_{i: \varepsilon_i \neq 0} \exp\left(- \frac{s^2}{2\varepsilon_i^2} \right),$$

we have

$$\begin{aligned} \forall i \in I_p: \quad & |\langle \psi_i, \Phi c_{p,\sigma} \rangle| \geq \alpha_i (c_S - \mu \|c\|_1) - s \\ \forall i \notin I_p: \quad & |\langle \psi_i, \Phi c_{p,\sigma} \rangle| \leq \alpha_i (c_{S+1} + \mu \|c\|_1) + s. \end{aligned}$$

This means that as long as $\min_{i \in I_p} \alpha_i (c_S - \mu \|c\|_1) - s \geq \max_{i \notin I_p} \alpha_i (c_{S+1} + \mu \|c\|_1) + s$, which is for instance implied by setting $s := \frac{1}{2}(c_S - c_{S+1} - 2\mu \|c\|_1 - \frac{\varepsilon^2}{2})$, we have

$$\max_{|I|=S} \|\Psi_I^\star \Phi c_{p,\sigma}\|_1 = \|\Psi_{I_p}^\star \Phi c_{p,\sigma}\|_1.$$

We now use this result for the calculation of the expectation over σ in (16). For any permutation p we define the set

$$\Sigma_p := \bigcup_i \{\sigma \text{ s.t. } \omega_i |\langle z_i, \Phi c_{p,\sigma} \rangle| \geq s\}.$$

We then have

$$\mathbb{E}_\sigma\left(\max_{|I|=S}\|\Psi_I^\star\Phi c_{p,\sigma}\|_1\right) = \sum_{\sigma\in\Sigma_p}\mathbb{P}(\sigma)\cdot\max_{|I|=S}\|\Psi_I^\star\Phi c_{p,\sigma}\|_1 + \sum_{\sigma\notin\Sigma_p}\mathbb{P}(\sigma)\cdot\|\Psi_{I_p}^\star\Phi c_{p,\sigma}\|_1$$
$$= \sum_{\sigma\in\Sigma_p}\mathbb{P}(\sigma)\cdot\left(\max_{|I|=S}\|\Psi_I^\star\Phi c_{p,\sigma}\|_1 - \|\Psi_{I_p}^\star\Phi c_{p,\sigma}\|_1\right) + \mathbb{E}_\sigma\left(\|\Psi_{I_p}^\star\Phi c_{p,\sigma}\|_1\right). \quad (17)$$

To estimate the sum over Σ_p, note that we have the following bounds:

$$|\langle\psi_i,\Phi c_{p,\sigma}\rangle| = |\alpha_i\langle\phi_i,\Phi c_{p,\sigma}\rangle + \omega_i\langle z_i,\Phi c_{p,\sigma}\rangle| \begin{cases} \leq (1-\frac{\varepsilon^2}{2})|\langle\phi_i,\Phi c_{p,\sigma}\rangle| + \varepsilon\sqrt{B} \\ \geq (1-\frac{\varepsilon^2}{2})|\langle\phi_i,\Phi c_{p,\sigma}\rangle| - \varepsilon\sqrt{B} \end{cases},$$

leading to

$$\max_{|I|=S}\|\Psi_I^\star\Phi c_{p,\sigma}\|_1 \leq (1-\frac{\varepsilon^2}{2})\max_{|I|=S}\|\Phi_I^\star\Phi c_{p,\sigma}\|_1 + S\cdot\varepsilon\sqrt{B} = (1-\frac{\varepsilon^2}{2})\|\Phi_{I_p}^\star\Phi c_{p,\sigma}\|_1 + S\cdot\varepsilon\sqrt{B}$$
$$\|\Psi_{I_p}^\star\Phi c_{p,\sigma}\|_1 \geq (1-\frac{\varepsilon^2}{2})\|\Phi_{I_p}^\star\Phi c_{p,\sigma}\|_1 - S\cdot\varepsilon\sqrt{B}.$$

Substituting these estimates into (17) we get

$$\mathbb{E}_\sigma\left(\max_{|I|=S}\|\Psi_I^\star\Phi c_{p,\sigma}\|_1\right) \leq \sum_{\sigma\in\Sigma_p}\mathbb{P}(\sigma)\cdot 2\varepsilon S\sqrt{B} + \mathbb{E}_\sigma\left(\|\Psi_{I_p}^\star\Phi c_{p,\sigma}\|_1\right)$$
$$\leq \eta\cdot 2\varepsilon S\sqrt{B} + \mathbb{E}_\sigma\left(\|\Psi_{I_p}^\star\Phi c_{p,\sigma}\|_1\right).$$

Next we calculate $\mathbb{E}_\sigma\left(\|\Psi_{I_p}^\star\Phi c_{p,\sigma}\|_1\right)$:

$$\mathbb{E}_\sigma\left(\|\Psi_{I_p}^\star\Phi c_{p,\sigma}\|_1\right) = \mathbb{E}_\sigma\left(\sum_{i\in I_p}|\langle\psi_i,\Phi c_{p,\sigma}\rangle|\right)$$
$$= \mathbb{E}_\sigma\left(\sum_{i\in I_p}\left|\alpha_i c_{p(i)} + \sigma_i\langle\alpha_i\phi_i+\omega_i z_i, \sum_{j\neq i}\sigma_j c_{p(j)}\phi_j\rangle\right|\right) = \sum_{i\in I_p}\alpha_i c_{p(i)}, \quad (18)$$

where have used that $\varepsilon \leq ((1-\frac{\varepsilon^2}{2})c_S - \mu\|c\|_1)/\sqrt{B}$ guarantees the expression within absolute values in (18) to always be positive. Collecting all these results we arrive at the following estimate for the value of the objective function at Ψ:

$$\mathbb{E}_y\left(\max_{|I|=S}\|\Psi_I^\star y\|_1\right) = \mathbb{E}_p\mathbb{E}_\sigma\left(\max_{|I|=S}\|\Psi_I^\star\Phi c_{p,\sigma}\|_1\right)$$
$$\leq \mathbb{E}_p\left(4\varepsilon S\sqrt{B}\sum_{i:\varepsilon_i\neq 0}\exp\left(-\frac{(c_S - c_{S+1} - 2\mu\|c\|_1 - \frac{\varepsilon^2}{2})^2}{8\varepsilon_i^2}\right) + \sum_{i\in I_p}\alpha_i c_{p(i)}\right)$$
$$\leq 4\varepsilon S\sqrt{B}\sum_{i:\varepsilon_i\neq 0}\exp\left(-\frac{(c_S - c_{S+1} - 2\mu\|c\|_1 - \frac{\varepsilon^2}{2})^2}{8\varepsilon_i^2}\right) + \frac{c_1+\ldots+c_S}{K}\sum_i\alpha_i.$$

Finally we are able to compare the expectation at the original dictionary to that at an ε-perturbation. Remembering that $\alpha_i = 1 - \frac{\varepsilon_i^2}{2}$, we get

$$\begin{aligned}
&\mathbb{E}_y\left(\max_{|I|=S}\|\Phi_I^\star y\|_1\right) - \mathbb{E}_y\left(\max_{|I|=S}\|\Psi_I^\star y\|_1\right) \\
&\qquad \geq \frac{c_1+\ldots+c_S}{K}\sum_i \frac{\varepsilon_i^2}{2} - 4\varepsilon S\sqrt{B}\sum_{i:\varepsilon_i\neq 0}\exp\left(-\frac{(c_S - c_{S+1} - 2\mu\|c\|_1 - \frac{\varepsilon^2}{2})^2}{8\varepsilon_i^2}\right) \\
&\qquad \geq \varepsilon^2\frac{c_1+\ldots+c_S}{2K} - 4\varepsilon SK\sqrt{B}\exp\left(-\frac{(c_S - c_{S+1} - 2\mu\|c\|_1 - \frac{\varepsilon^2}{2})^2}{8\varepsilon^2}\right).
\end{aligned}$$

Thus to have a local maximum at the original dictionary we need that

$$\varepsilon > \frac{8SK^2\sqrt{B}}{c_1+\ldots+c_S}\exp\left(-\frac{(c_S - c_{S+1} - 2\mu\|c\|_1 - \frac{\varepsilon^2}{2})^2}{8\varepsilon^2}\right),$$

and all that remains to be shown is that this is implied by (15). Since $K \geq 2$, (15) implies that $\frac{\varepsilon^2}{2} < \frac{c_S - c_{S+1} - 2\mu\|c\|_1}{2(1+3\sqrt{\log 96})^2} \leq \frac{c_S - c_{S+1} - 2\mu\|c\|_1}{100}$ and it suffices to show that (15) further implies

$$\varepsilon > \frac{8SK^2\sqrt{B}}{c_1+\ldots+c_S}\exp\left(-\frac{(c_S - c_{S+1} - 2\mu\|c\|_1)^2\cdot 99^2}{8\varepsilon^2\cdot 100^2}\right). \tag{19}$$

Applying Lemma A.3 by Schnass (2014), which says that for $a, b, \varepsilon > 0$,

$$\varepsilon \leq \frac{4b}{1+\sqrt{1+16\log(\frac{a}{b})}} \quad \text{implies that} \quad a\exp\left(\frac{-b^2}{\varepsilon^2}\right) < \varepsilon,$$

to the situation at hand, where $a = \frac{8SK^2\sqrt{B}}{c_1+\ldots+c_S}$ and $b = \frac{(c_S - c_{S+1} - 2\mu\|c\|_1)\cdot 99}{\sqrt{8}\cdot 100}$, we get that (19) is ensured by

$$\varepsilon < \frac{c_S - c_{S+1} - 2\mu\|c\|_1}{\sqrt{8}\cdot\frac{25}{99}\left(1+\sqrt{16\log\left(\frac{8\sqrt{8}\cdot\frac{100}{99}e^{1/16}SK^2\sqrt{B}}{(c_S - c_{S+1} - 2\mu\|c\|_1)(c_1+\ldots+c_S)}\right)}\right)},$$

which simplifies to (15). ■

Proof [of Theorem 2]
Using the symmetry of ν, our strategy is to reduce the general to the simple coefficient model. Let c denote the mapping that assigns to each $x \in S^{K-1}$ the non increasing rearrangement of the absolute values of its components, that is $c_i(x) = |x_{p(i)}|$ for a permutation p such that $c_1(x) \geq c_2(x) \geq \ldots \geq c_K(x) \geq 0$. Then the mapping c together with the probability measure ν on S^{K-1} induces a pull-back probability measure ν_c on $c(S^{K-1})$, by $\nu_c(\Omega) := \nu(c^{-1}(\Omega))$ for any measurable set $\Omega \subseteq c(S^{K-1})$. With the help of this new measure

we can rewrite the expectations we need to calculate as

$$\mathbb{E}_y\left(\max_{|I|=S}\|\Phi_I^\star y\|_1\right) = \mathbb{E}_x\left(\max_{|I|=S}\|\Phi_I^\star \Phi x\|_1\right)$$
$$= \int_x \max_{|I|=S}\|\Phi_I^\star \Phi x\|_1 d\nu = \int_{c(x)} \mathbb{E}_p\mathbb{E}_\sigma\left(\max_{|I|=S}\|\Phi^\star\Phi c_{p,\sigma}(x)\|_1\right) d\nu_c.$$

The expectation inside the integral should seem familiar. Indeed we have calculated it already in the proof of Proposition 6 for $c(x)$ a fixed decaying sequence satisfying $c_S(x) > c_{S+1}(x) + 2\mu\|x\|_1$. Since this property is satisfied almost surely we have

$$\mathbb{E}_y\left(\max_{|I|=S}\|\Phi_I^\star y\|_1\right) = \int_{c(x)} \mathbb{E}_p\mathbb{E}_\sigma\left(\max_{|I|=S}\|\Phi^\star\Phi c_{p,\sigma}(x)\|_1\right) d\nu_c$$
$$= \int_{c(x)} c_1(x) + \ldots + c_S(x) d\nu_c := \bar{c}_1 + \ldots + \bar{c}_S.$$

For the expectation of a perturbed dictionary Ψ we get in analogy

$$\mathbb{E}_y\left(\max_{|I|=S}\|\Psi_I^\star y\|_1\right) = \int_{c(x)} \mathbb{E}_p\mathbb{E}_\sigma\left(\max_{|I|=S}\|\Psi^\star\Phi c_{p,\sigma}(x)\|_1\right) d\nu_c$$
$$\leq \int_{c(x)} \eta(x) + (c_1(x) + \ldots + c_S(x))\,\frac{1}{K}\sum_i \alpha_i\, d\nu_c,$$

where

$$\eta(x) := 4\varepsilon S\sqrt{B}\sum_{i:\varepsilon_i\neq 0}\exp\left(-\frac{(c_S(x) - c_{S+1}(x) - 2\mu\|x\|_1 - \frac{\varepsilon^2}{2})^2}{8\varepsilon_i^2}\right).$$

Since $c_S(x) - c_{S+1}(x) - 2\mu\|x\|_1 \geq \beta$ almost surely we have

$$\eta(x) \leq 4\varepsilon S\sqrt{B}\sum_{i:\varepsilon_i\neq 0}\exp\left(-\frac{(\beta - \frac{\varepsilon^2}{2})^2}{8\varepsilon_i^2}\right) := \eta_\beta,$$

almost surely and therefore

$$\mathbb{E}_y\left(\max_{|I|=S}\|\Psi_I^\star y\|_1\right) \leq \eta_\beta + (\bar{c}_1 + \ldots + \bar{c}_S)\,\frac{1}{K}\sum_i \alpha_i.$$

Following the same argument as in the proof of Proposition 6 we see that $\mathbb{E}_y\left(\max_{|I|=S}\|\Phi_I^\star y\|_1\right) > \mathbb{E}_y\left(\max_{|I|=S}\|\Psi_I^\star y\|_1\right)$ as soon as

$$\varepsilon < \frac{\beta}{1 + 3\sqrt{\log\left(\frac{25K^2S\sqrt{B}}{\beta(\bar{c}_1+\ldots+\bar{c}_S)}\right)}}.$$

■

A.2 Proof of Theorem 3

Again we first reformulate and prove the theorem for the case of a symmetric coefficient distribution based on one sequence and then extend it with an integration argument.

Proposition 7 *Let Φ be a unit norm frame with frame constants $A \leq B$ and coherence μ. Let $x \in \mathbb{R}^K$ be a random permutation of a sequence c, where $c_1 \geq c_2 \geq c_3 \ldots \geq c_K \geq 0$ and $\|c\|_2 = 1$, provided with random $\pm$ signs, that is $x = c_{p,\sigma}$ with probability $\mathbb{P}(p,\sigma) = (2^K K!)^{-1}$. Further let $r = (r(1) \ldots r(d))$ be a centred random subgaussian noise-vector with parameter ρ and assume that the signals are generated according to the noisy signal model in (9). If we have*

$$\max\{\mu, \rho\} \leq \frac{c_S - c_{S+1}}{\sqrt{72(\log a + \log\log a)}} \quad \textit{for} \quad a = \frac{112K^2 S(\sqrt{B}+1)}{C_r(c_S - c_{S+1})(c_1 + \ldots + c_S)}, \tag{20}$$

where $C_r = \mathbb{E}_r\left((1+\|r\|_2^2)^{-1/2}\right)$, then there is a local maximum of (5) at $\tilde{\Psi}$ satisfying

$$d(\tilde{\Psi}, \Phi) \leq \frac{12SK^2\sqrt{B}}{C_r(c_1 + \ldots + c_S)} \exp\left(\frac{-(c_S - c_{S+1})^2}{72\max\{\mu^2, \rho^2\}}\right).$$

Proof To prove the proposition we digress from the conventional scheme of first calculating the expectation of our objective function for both the original and a perturbed dictionary and then comparing and instead bound the difference of the expectations directly.

$$\begin{aligned}
&\mathbb{E}_y\left(\max_{|I|=S} \|\Phi_I^\star y\|_1\right) - \mathbb{E}_y\left(\max_{|I|=S} \|\Psi_I^\star y\|_1\right) \\
&= \mathbb{E}_{p,\sigma,r}\left(\max_{|I|=S}\left\|\frac{\Phi_I^\star(\Phi c_{p,\sigma}+r)}{\sqrt{1+\|r\|_2^2}}\right\|_1 - \max_{|I|=S}\left\|\frac{\Psi_I^\star(\Phi c_{p,\sigma}+r)}{\sqrt{1+\|r\|_2^2}}\right\|_1\right) \\
&= \mathbb{E}_{p,\sigma,r}\left(\frac{\max_{|I|=S}\|\Phi_I^\star(\Phi c_{p,\sigma}+r)\|_1 - \max_{|I|=S}\|\Psi_I^\star(\Phi c_{p,\sigma}+r)\|_1}{\sqrt{1+\|r\|_2^2}}\right) := \mathbb{E}_{p,\sigma,r}(\Delta_{p,\sigma,r})
\end{aligned}$$

Again our strategy is to show that for a fixed p for most σ and r the maximal response of both the original dictionary and the perturbation is attained at I_p. The expressions we therefore need to lower (upper) bound for $i \in I_p$ ($i \notin I_p$) are

$$\begin{aligned}
|\langle \phi_i, \Phi c_{p,\sigma} + r\rangle| &= \Big|\sigma_i c_{p(i)} + \sum_{j\neq i} \sigma_j c_{p(j)}\langle\phi_i,\phi_j\rangle + \langle\phi_i, r\rangle\Big|, \\
|\langle \psi_i, \Phi c_{p,\sigma} + r\rangle| &= \Big|\alpha_i\sigma_i c_{p(i)} + \alpha_i\sum_{j\neq i} \sigma_j c_{p(j)}\langle\phi_i,\phi_j\rangle + \omega_i\langle z_i, \Phi c_{p,\sigma}\rangle + \langle\psi_i, r\rangle\Big|.
\end{aligned}$$

However, instead of using a worst case estimate for the gap between the responses of the original dictionary inside and outside I_p, we now make use of the fact that for most sign sequences we have a gap size of order $c_S - c_{S+1}$. This means that as soon as $|\sum_{j\neq i}\sigma_j c_{p(j)}\langle\phi_i,\phi_j\rangle|$, $\omega_i|\langle z_i, \Phi c_{p,\sigma}\rangle|$ and the noise related terms $|\langle\phi_i, r\rangle|$ and $|\langle\psi_i, r\rangle|$ are of order $(c_S - c_{S+1})$ the maximal response of both the original dictionary and the perturbation is attained at I_p. In

particular, defining the sets

$$\Sigma_p := \bigcup_i \left\{ \sigma \text{ s.t. } \left| \sum_{j\neq i} \sigma_j c_{p(j)} \langle \phi_i, \phi_j \rangle \right| \geq \frac{c_S - c_{S+1}}{6} \text{ or } \omega_i |\langle z_i, \Phi c_{p,\sigma} \rangle| \geq \frac{c_S - c_{S+1} - \frac{3\varepsilon^2}{2}}{6} \right\},$$

for a fixed permutation p and

$$R := \bigcup_i \left\{ r \text{ s.t. } |\langle \phi_i, r \rangle| \geq \frac{c_S - c_{S+1}}{3} \text{ or } |\langle \psi_i, r \rangle| \geq \frac{c_S - c_{S+1}}{6} \right\},$$

we see that both maxima are attained at I_p as long as $\sigma \notin \Sigma_p$ and $r \notin R$. Using Hoeffding's inequality we get that

$$\mathbb{P}\left(\left| \sum_{j\neq i} \sigma_j c_{p(j)} \langle \phi_i, \phi_j \rangle \right| > t \right) \leq 2 \exp\left(\frac{-t^2}{2\sum_{j\neq i} c_{p(j)}^2 |\langle \phi_i, \phi_j \rangle|^2} \right) \leq 2 \exp\left(\frac{-t^2}{2\mu^2} \right),$$

while from the proof of Proposition 6 we know that for $\varepsilon_i \neq 0$ we have $\mathbb{P}(\omega_i |\langle z_i, \Phi c_{p,\sigma} \rangle| \geq s) \leq 2\exp\left(-\frac{s^2}{2\varepsilon_i^2}\right)$. Setting $t = (c_S - c_{S+1})/6$, $s = (c_S - c_{S+1} - \frac{3\varepsilon^2}{2})/6$ and using a union bound then leads to

$$\mathbb{P}(\Sigma_p) \leq 2K \exp\left(-\frac{\left(c_S - c_{S+1} - \frac{3\varepsilon^2}{2}\right)^2}{72\varepsilon^2} \right) + 2K \exp\left(\frac{-(c_S - c_{S+1})^2}{72\mu^2} \right). \tag{21}$$

Since the $r(i)$ are subgaussian with parameter ρ we have for any $v = (v_1 \dots v_d)$ and $t \geq 0$, $\mathbb{P}(|\langle v, r \rangle| \geq t) \leq \exp\left(-\frac{t^2}{2\rho^2 \|v\|_2^2}\right)$ (Vershynin, 2012). Taking a union bound over all ϕ_i, ψ_i with the corresponding choice for t then leads to the estimate

$$\mathbb{P}(R) \leq 2K \exp\left(-\frac{\left(c_S - c_{S+1}\right)^2}{72\rho^2} \right) + 2K \exp\left(-\frac{\left(c_S - c_{S+1}\right)^2}{18\rho^2} \right). \tag{22}$$

We now split the expectations over the sign and noise patterns for a fixed p to get

$$\begin{aligned} \mathbb{E}_\sigma \mathbb{E}_r(\Delta_{p,\sigma,r}) &= \mathbb{E}_\sigma \left(\int_{r\notin R} \Delta_{p,\sigma,r} d\nu_r \right) + \mathbb{E}_\sigma \left(\int_{r\in R} \Delta_{p,\sigma,r} d\nu_r \right) \\ &= \sum_{\sigma\notin\Sigma_p} \mathbb{P}(\sigma) \int_{r\notin R} \Delta_{p,\sigma,r} d\nu_r + \sum_{\sigma\in\Sigma_p} \mathbb{P}(\sigma) \int_{r\notin R} \Delta_{p,\sigma,r} d\nu_r \\ &\qquad + \mathbb{E}_\sigma \left(\int_{r\in R} \Delta_{p,\sigma,r} d\nu_r \right). \end{aligned} \tag{23}$$

Next note that Σ_p is symmetric in the sense that we either have $(\sigma_1, \dots, \pm\sigma_i, \dots, \sigma_K) \in \Sigma_p$ or $(\sigma_1, \dots, \pm\sigma_i, \dots, \sigma_K) \notin \Sigma_p$. Thus we get for the first term in (23),

$$\begin{aligned} \sum_{\sigma\notin\Sigma_p} \mathbb{P}(\sigma) \int_{r\notin R} \Delta_{p,\sigma,r} d\nu_r &= \int_{r\notin R} \sum_{\sigma\notin\Sigma_p} \mathbb{P}(\sigma) \left(\frac{\left\| \Phi_{I_p}^\star (\Phi c_{p,\sigma} + r) \right\|_1 - \left\| \Psi_{I_p}^\star (\Phi c_{p,\sigma} + r) \right\|_1}{\sqrt{1 + \|r\|_2^2}} \right) d\nu_r \\ &= \int_{r\notin R} \sum_{\sigma\notin\Sigma_p} \mathbb{P}(\sigma) \left(\frac{\sum_{i\in I_p} c_{p(i)} \frac{\varepsilon_i^2}{2}}{\sqrt{1 + \|r\|_2^2}} \right) d\nu_r. \end{aligned}$$

To bound the last two terms in (23) we first find an upper bound formax$_{|I|=S} \|\Psi_I^\star(\Phi c_{p,\sigma} + r)\|_1$:

$$\begin{aligned}
\max_{|I|=S} \|\Psi_I^\star(\Phi c_{p,\sigma} + r)\|_1 &= \max_{|I|=S} \sum_{i\in I} |\langle \alpha_i \phi_i + \omega_i z_i, \Phi c_{p,\sigma} + r)\rangle| \\
&\le \max_{|I|=S} \sum_{i\in I} \big(1 - \frac{\varepsilon_i^2}{2}\big)|\langle \phi_i, \Phi c_{p,\sigma} + r)\rangle| + \varepsilon_i \|\Phi c_{p,\sigma} + r\|_2 \\
&\le \max_{|I|=S} \sum_{i\in I} \big(1 - \frac{\varepsilon^2}{2}\big)|\langle \phi_i, \Phi c_{p,\sigma} + r)\rangle| + \varepsilon\big(\sqrt{B} + \|r\|_2\big) \\
&= \big(1 - \frac{\varepsilon^2}{2}\big) \max_{|I|=S} \|\Phi_I^\star(\Phi c_{p,\sigma} + r)\|_1 + \varepsilon S\big(\sqrt{B} + \|r\|_2\big).
\end{aligned}$$

This then leads to the following lower bound for $\Delta_{p,\sigma,r}$:

$$\begin{aligned}
\Delta_{p,\sigma,r} &\ge (1 + \|r\|_2^2)^{-1/2} \left(\max_{|I|=S} \|\Phi_I^\star(\Phi c_{p,\sigma} + r)\|_1 \frac{\varepsilon^2}{2} - \varepsilon S\big(\sqrt{B} + \|r\|_2\big) \right) \\
&\ge (1 + \|r\|_2^2)^{-1/2} \left(\|\Phi_{I_p}^\star(\Phi c_{p,\sigma} + r)\|_1 \frac{\varepsilon^2}{2} - \varepsilon S\big(\sqrt{B} + \|r\|_2\big) \right).
\end{aligned}$$

Using again the symmetry of Σ_p we have

$$\begin{aligned}
\sum_{\sigma\in\Sigma_p} \mathbb{P}(\sigma) \int_{r\notin R} \Delta_{p,\sigma,r} d\nu_r &\ge \int_{r\notin R} \sum_{\sigma\in\Sigma_p} \mathbb{P}(\sigma) \frac{\|\Phi_{I_p}^\star(\Phi c_{p,\sigma} + r)\|_1 \frac{\varepsilon^2}{2} - \varepsilon S\big(\sqrt{B} + \|r\|_2\big)}{\sqrt{1 + \|r\|_2^2}} d\nu_r \\
&\ge \int_{r\notin R} \sum_{\sigma\in\Sigma_p} \mathbb{P}(\sigma) \left(\frac{\sum_{i\in I_p} c_{p(i)} \frac{\varepsilon^2}{2}}{\sqrt{1 + \|r\|_2^2}} - \varepsilon S\big(\sqrt{B} + 1\big) \right) d\nu_r \\
&\ge \int_{r\notin R} \sum_{\sigma\in\Sigma_p} \mathbb{P}(\sigma) \left(\frac{\sum_{i\in I_p} c_{p(i)} \frac{\varepsilon_i^2}{2}}{\sqrt{1 + \|r\|_2^2}} - \varepsilon S\big(\sqrt{B} + 1\big) \right) d\nu_r,
\end{aligned}$$

and similarly

$$\begin{aligned}
\mathbb{E}_\sigma \left(\int_{r\in R} \Delta_{p,\sigma,r} d\nu_r \right) &\ge \int_{r\in R} \mathbb{E}_\sigma \left(\frac{\|\Phi_{I_p}^\star(\Phi c_{p,\sigma} + r)\|_1 \frac{\varepsilon^2}{2} - \varepsilon S\big(\sqrt{B} + \|r\|_2\big)}{\sqrt{1 + \|r\|_2^2}} \right) d\nu_r \\
&\ge \int_{r\in R} \frac{\sum_{i\in I_p} c_{p(i)} \frac{\varepsilon_i^2}{2}}{\sqrt{1 + \|r\|_2^2}} - \varepsilon S\big(\sqrt{B} + 1\big) d\nu_r.
\end{aligned}$$

Resubstituting into (23) we get

$$\begin{aligned}\mathbb{E}_\sigma\mathbb{E}_r(\Delta_{p,\sigma,r}) &\geq \int_{r\in R}\frac{\sum_{i\in I_p} c_{p(i)}\frac{\varepsilon_i^2}{2}}{\sqrt{1+\|r\|_2^2}} - \varepsilon S\big(\sqrt{B}+1\big)d\nu_r \\ &\qquad + \int_{r\notin R}\sum_{\sigma\in\Sigma_p}\mathbb{P}(\sigma)\left(\frac{\sum_{i\in I_p} c_{p(i)}\frac{\varepsilon_i^2}{2}}{\sqrt{1+\|r\|_2^2}} - \varepsilon S\big(\sqrt{B}+1\big)\right)d\nu_r \\ &\qquad\qquad + \int_{r\notin R}\sum_{\sigma\notin\Sigma_p}\mathbb{P}(\sigma)\left(\frac{\sum_{i\in I_p} c_{p(i)}\frac{\varepsilon_i^2}{2}}{\sqrt{1+\|r\|_2^2}}\right)d\nu_r \\ &\geq \int_r \frac{\sum_{i\in I_p} c_{p(i)}\frac{\varepsilon_i^2}{2}}{\sqrt{1+\|r\|_2^2}}d\nu_r - \varepsilon S\big(\sqrt{B}+1\big)\cdot\big(P(R)+P(\Sigma_p)\big). \qquad (24)\end{aligned}$$

Taking the expectation over the permutations then yields

$$\begin{aligned}\mathbb{E}_{p,\sigma,r}(\Delta_{p,\sigma,r}) &\geq \mathbb{E}_r\mathbb{E}_p\left(\frac{\sum_{i\in I_p} c_{p(i)}\frac{\varepsilon_i^2}{2}}{\sqrt{1+\|r\|_2^2}}\right) - \varepsilon S\big(\sqrt{B}+1\big)\cdot\big(\mathbb{P}(R)+\mathbb{E}_p\mathbb{P}(\Sigma_p)\big) \\ &\geq \mathbb{E}_r\left(\frac{1}{\sqrt{1+\|r\|_2^2}}\right)\frac{c_1+\ldots+c_S}{2K}\sum_i\varepsilon_i^2 - \varepsilon S\big(\sqrt{B}+1\big)\cdot\big(\mathbb{P}(R)+\mathbb{E}_p\mathbb{P}(\Sigma_p)\big).\end{aligned}$$

Using the probability estimates from (21)/(22) we see that $\mathbb{E}_{p,\sigma,r}(\Delta_{p,\sigma,r})>0$ is implied by

$$\varepsilon \geq \frac{4SK^2\big(\sqrt{B}+1\big)}{C_r\gamma}\left(\exp\left(\frac{-\big(\beta-\frac{3\varepsilon^2}{2}\big)^2}{72\varepsilon^2}\right)+\exp\left(\frac{-\beta^2}{72\mu^2}\right)+\exp\left(\frac{-\beta^2}{72\rho^2}\right)+\exp\left(\frac{-\beta^2}{18\rho^2}\right)\right),$$

where we have used the abbreviations $\gamma = c_1+\ldots+c_S$, $\beta = c_S - c_{S+1}$ and $C_r = \mathbb{E}_r\left((1+\|r\|_2^2)^{-1/2}\right)$. We now proceed by splitting the above condition. We define $\varepsilon_{\min}$ by asking that

$$\frac{\varepsilon}{3} \geq \frac{4SK^2\big(\sqrt{B}+1\big)}{C_r\gamma}\exp\left(-\frac{\beta^2}{72\max\{\mu^2,\rho^2\}}\right) := \frac{\varepsilon_{\min}}{3}$$

and $\varepsilon_{\max}$ implicitly by asking that

$$\frac{\varepsilon}{3}-\frac{\varepsilon^4}{81} \geq \frac{4SK^2\big(\sqrt{B}+1\big)}{C_r\gamma}\exp\left(-\frac{\big(\beta-\frac{3\varepsilon^2}{2}\big)^2}{72\varepsilon^2}\right).$$

Following the line of argument in the proof of Proposition 6 we see that the above condition is guaranteed as soon as

$$\varepsilon \leq \frac{\beta}{\frac{5}{2}+9\sqrt{\log\left(\frac{112K^2S(\sqrt{B}+1)}{C_r\beta\gamma}\right)}} := \varepsilon_{\max}.$$

The statement follows from making sure that $\varepsilon_{\min} < \varepsilon_{\max}$. ■

Proof [of Theorem 3] Using the pull-back probability measure ν_c we can write

$$\mathbb{E}_y\left(\max_{|I|=S}\|\Phi_I^\star y\|_1\right) - \mathbb{E}_y\left(\max_{|I|=S}\|\Psi_I^\star y\|_1\right) = \int_{c(x)} \mathbb{E}_{p,\sigma,r}\left(\Delta_{p,\sigma,r,c(x)}\right) d\nu_c,$$

where $\Delta_{p,\sigma,r,c(x)}$ is defined analogue to $\Delta_{p,\sigma,r}$ in the last proof, that is replacing c by $c(x)$. The statement follows from employing the lower estimate for $\mathbb{E}_{p,\sigma,r}\left(\Delta_{p,\sigma,r,c(x)}\right)$ from (24) and replacing $c_1 + \ldots + c_S$ by $\bar{c}_1 + \ldots + \bar{c}_S$ resp. $c_S - c_{S+1}$ by its lower bound β in the proof of Proposition 7. ■

A.3 Proof of Theorems 4 and 5

Since the proofs of Theorems 4 and 5 are conceptually equivalent we will combine them into one and just split the argument for the inevitable juggling of constants.

Proof As outlined in the proof idea we need a Lipschitz property for the mapping $\Psi \to \frac{1}{N}\sum_{n=1}^N \max_{|I|=S}\|\Psi_I^\star y_n\|_1$ for both signal models, the concentration of the sum around its expectation for a δ net covering the space of all admissible dictionaries close to Φ and a triangle inequality argument to get to the final statement.

To show the Lipschitz property we use a reverse triangle inequality:

$$\begin{aligned}\left|\max_{|I|=S}\|\Psi_I^\star y_n\|_1 - \max_{|I|=S}\|\bar{\Psi}_I^\star y_n\|_1\right| &= \left|\max_{|I|=S}\|\bar{\Psi}_I^\star y_n - (\bar{\Psi}_I^\star - \Psi_I^\star)y_n\|_1 - \max_{|I|=S}\|\bar{\Psi}_I^\star y_n\|_1\right| \\ &\leq \max_{|I|=S}\|(\bar{\Psi}_I^\star - \Psi_I^\star)y_n\|_1 \\ &\leq S\max_k \|\psi_k - \bar{\psi}_k\|_2 \|y_n\|_2 \\ &\leq d(\Psi,\bar{\Psi})S\big(\sqrt{B}+1\big).\end{aligned}$$

Note that for the noise-free signal model we can replace $\big(\sqrt{B}+1\big)$ by $\sqrt{B}$ in the last expression. By averaging over n we get that the mapping in question is Lipschitz with constant $S\big(\sqrt{B}+1\big)$ in the noisy and $S\sqrt{B}$ in the noise-free case, that is

$$\left|\frac{1}{N}\sum_{n=1}^N \max_{|I|=S}\|\Psi_I^\star y_n\|_1 - \frac{1}{N}\sum_{n=1}^N \max_{|I|=S}\|\bar{\Psi}_I^\star y_n\|_1\right| \leq d(\Psi,\bar{\Psi})S\big(\sqrt{B}+1\big).$$

To show that the averaged sums concentrate around their expectations we use our favourite tool Hoeffding's inequality. Set $X_n = \max_{|I|=S}\|\Phi_I^\star y_n\|_1 - \max_{|I|=S}\|\Psi_I^\star y_n\|_1$, then we have $|X_n| \leq \varepsilon S\big(\sqrt{B}+1\big)$, resp. $|X_n| \leq \varepsilon S\sqrt{B}$ in the noise-free case, and get the estimate

$$\begin{aligned}\mathbb{P}\left(\left|\frac{1}{N}\sum_{n=1}^N\left(\max_{|I|=S}\|\Phi_I^\star y_n\|_1 - \max_{|I|=S}\|\Psi_I^\star y_n\|_1\right) - \mathbb{E}\left(\max_{|I|=S}\|\Phi_I^\star y_1\|_1 - \max_{|I|=S}\|\Psi_I^\star y_1\|_1\right)\right| \geq 2t\right) \\ \leq 2\exp\left(\frac{-2Nt^2}{\varepsilon^2 S^2\big(\sqrt{B}+1\big)^2}\right).\end{aligned}$$

Next we need to choose a δ-net for all perturbations Ψ with $d(\Phi,\Psi)\leq\varepsilon_{\max}$, that is a finite set of perturbations $\mathcal{N}$ such that for every Ψ we can find $\bar{\Psi}\in\mathcal{N}$ with $d(\Psi,\bar{\Psi})\leq\delta$. Recalling the parameterization of all ε-perturbations from the proof of Proposition 6, we see that the space we need to cover is included in the product of K balls with radius $\varepsilon_{\max}$ in dimension d. For instance from Lemma 2 by Vershynin (2012) we know that for the d dimensional ball of radius $\varepsilon_{\max}$ we can find a δ-net $\mathcal{N}_d$ satisfying $\sharp\mathcal{N}_d\leq\left(\varepsilon_{\max}+\frac{2\varepsilon_{\max}}{\delta}\right)^d$, so for our space of ε-perturbations we can find a δ-net $\mathcal{N}$ satisfying

$$\sharp\mathcal{N}\leq\left(\varepsilon_{\max}+\frac{2\varepsilon_{\max}}{\delta}\right)^{Kd}\leq\left(\frac{3\varepsilon_{\max}}{\delta}\right)^{Kd}.$$

Taking a union bound we can now estimate the probability that we have concentration for all perturbations in the net as

$$\begin{aligned}\mathbb{P}\Bigg(\exists\Psi\in\mathcal{N}:\Bigg|\frac{1}{N}\sum_{n=1}^{N}\Big(\max_{|I|=S}\|\Phi_I^\star y_n\|_1-\max_{|I|=S}\|\Psi_I^\star y_n\|_1\Big)\\ -\mathbb{E}\left(\max_{|I|=S}\|\Phi_I^\star y_1\|_1-\max_{|I|=S}\|\Psi_I^\star y_1\|_1\right)\Bigg|\geq 2t\Bigg)\\ \leq\left(\frac{3\varepsilon_{\max}}{\delta}\right)^{Kd}2\exp\left(\frac{-2Nt^2}{\varepsilon_{\max}^2S^2\left(\sqrt{B}+1\right)^2}\right).\end{aligned}$$

Finally we are ready for the triangle inequality argument. For any Ψ with $d(\Psi,\Phi)=\varepsilon\leq\varepsilon_{\max}$ we can find $\bar{\Psi}\in\mathcal{N}$ with $d(\bar{\Psi},\Psi)\leq\delta$ and $d(\Phi,\bar{\Psi})=\bar{\varepsilon}$ and therefore get

$$\begin{aligned}&\frac{1}{N}\sum_{n=1}^{N}\max_{|I|=S}\|\Phi_I^\star y_n\|_1-\frac{1}{N}\sum_{n=1}^{N}\max_{|I|=S}\|\Psi_I^\star y_n\|_1\\ &=\frac{1}{N}\sum_{n=1}^{N}\max_{|I|=S}\|\Phi_I^\star y_n\|_1-\mathbb{E}\left(\max_{|I|=S}\|\Phi_I^\star y_1\|_1\right)+\mathbb{E}\left(\max_{|I|=S}\|\Phi_I^\star y_1\|_1\right)-\mathbb{E}\left(\max_{|I|=S}\|\bar{\Psi}_I^\star y_1\|_1\right)\\ &+\mathbb{E}\left(\max_{|I|=S}\|\bar{\Psi}_I^\star y_1\|_1\right)-\frac{1}{N}\sum_{n=1}^{N}\max_{|I|=S}\|\bar{\Psi}_I^\star y_n\|_1+\frac{1}{N}\sum_{n=1}^{N}\max_{|I|=S}\|\bar{\Psi}_I^\star y_n\|_1-\frac{1}{N}\sum_{n=1}^{N}\max_{|I|=S}\|\Psi_I^\star y_n\|_1\\ &\geq\mathbb{E}\left(\max_{|I|=S}\|\Phi_I^\star y_1\|_1\right)-\mathbb{E}\left(\max_{|I|=S}\|\bar{\Psi}_I^\star y_1\|_1\right)-2t-\delta S\left(\sqrt{B}+1\right).\end{aligned}$$

Depending on the signal model we now have to substitute the values for the asymptotic differences $\mathbb{E}\left(\max_{|I|=S}\|\Phi_I^\star y_1\|_1\right)-\mathbb{E}\left(\max_{|I|=S}\|\bar{\Psi}_I^\star y_1\|_1\right)$ calculated in the previous proofs. Under the conditions given in Theorem 4 we have,

$$\begin{aligned}&\frac{1}{N}\sum_{n=1}^{N}\max_{|I|=S}\|\Phi_I^\star y_n\|_1-\frac{1}{N}\sum_{n=1}^{N}\max_{|I|=S}\|\Psi_I^\star y_n\|_1\\ &\qquad\geq\bar{\varepsilon}^2\frac{\bar{c}_1+\ldots+\bar{c}_S}{2K}-4\bar{\varepsilon}SK\sqrt{B}\exp\left(-\frac{(\beta-\frac{\bar{\varepsilon}^2}{2})^2}{8\bar{\varepsilon}^2}\right)-2t-\delta S\sqrt{B}.\end{aligned}\tag{25}$$

To make sure that the above expression is larger than zero, we split it into two conditions. The first condition

$$\bar{\varepsilon}^2 \frac{\bar{c}_1 + \ldots + \bar{c}_S}{4K} > 4\bar{\varepsilon} S K \sqrt{B} \exp\left(-\frac{(\beta - \frac{\bar{\varepsilon}^2}{2})^2}{8\bar{\varepsilon}^2} \right)$$

is satisfied as soon as

$$\bar{\varepsilon} \leq \frac{\beta}{\frac{25\sqrt{8}}{99}\left(1 + 4\sqrt{\log\left(\frac{50K^2 S\sqrt{B}}{\beta(\bar{c}_1 + \ldots + \bar{c}_S)}\right)}\right)}.$$

To concretise the second condition

$$\bar{\varepsilon}^2 \frac{\bar{c}_1 + \ldots + \bar{c}_S}{4K} \geq 2t + \delta S \sqrt{B},$$

we choose $t = \tilde{\varepsilon}^2 \frac{\bar{c}_1 + \ldots + \bar{c}_S}{16K}$ and $\delta = \tilde{\varepsilon}^2 \frac{\bar{c}_1 + \ldots + \bar{c}_S}{8KS\sqrt{B}}$ to arrive at $\bar{\varepsilon} \geq 2\tilde{\varepsilon}$. Given that $\bar{\varepsilon}$ differs at most by δ from ε we see that (25) is larger than zero except with probability

$$2\exp\left(-\frac{N\tilde{\varepsilon}^4 (\bar{c}_1 + \ldots + \bar{c}_S)^2}{128\varepsilon_{\max}^2 S^2 K^2 B} + Kd \log\left(\frac{24\varepsilon_{\max} KS\sqrt{B}}{\tilde{\varepsilon}^2(\bar{c}_1 + \ldots + \bar{c}_S)} \right) \right),$$

as long as

$$\varepsilon_{\min} := \tilde{\varepsilon} + \frac{\tilde{\varepsilon}^2}{8K} \leq \varepsilon \leq \varepsilon_{\max} \leq \frac{\beta}{\frac{25\sqrt{8}}{99}\left(1 + 4\sqrt{\log\left(\frac{50K^2 S\sqrt{B}}{\beta(\bar{c}_1 + \ldots + \bar{c}_S)}\right)}\right)} - \frac{\tilde{\varepsilon}^2}{8K}. \tag{26}$$

While for the asymptotic results we tried to make $\varepsilon_{\max}$ as large as possible to indicate how large the basin of attraction could be, for the finite sample size results we want it as small as possible in order to keep the sampling complexity small and therefore choose $\varepsilon_{\max} = \varepsilon_{\min}$. The statement then follows from making sure that the right most inequality in (26) is satisfied and simplifications.
In case of the noisy signal model, that is under the conditions given in Theorem 5, we have

$$\begin{aligned}
&\frac{1}{N}\sum_{n=1}^{N} \max_{|I|=S} \|\Phi_I^\star y_n\|_1 - \frac{1}{N}\sum_{n=1}^{N} \max_{|I|=S} \|\Psi_I^\star y_n\|_1 \\
&\quad \geq \bar{\varepsilon}^2 \frac{\bar{c}_1 + \ldots + \bar{c}_S}{2C_r K} - 2t - \delta S\big(\sqrt{B} + 1\big) - 2\bar{\varepsilon} SK\big(\sqrt{B} + 1\big) \cdot \\
&\qquad \cdot \left(\exp\left(\frac{-\big(\beta - \frac{3\bar{\varepsilon}^2}{2}\big)^2}{72\bar{\varepsilon}^2} \right) + \exp\left(\frac{-\beta^2}{72\mu^2} \right) + \exp\left(\frac{-\beta^2}{72\rho^2} \right) + \exp\left(-\frac{\beta^2}{18\rho^2} \right) \right).
\end{aligned} \tag{27}$$

Splitting equally gives us four conditions:

$$\begin{aligned}
\bar{\varepsilon} &\geq \frac{16C_rSK^2\big(\sqrt{B}+1\big)}{\bar{c}_1+\ldots+\bar{c}_S}\exp\left(-\frac{\beta^2}{72\mu^2}\right) := \varepsilon_\mu,\\
\bar{\varepsilon} &\geq \frac{16C_rSK^2\big(\sqrt{B}+1\big)}{\bar{c}_1+\ldots+\bar{c}_S}\exp\left(-\frac{\beta^2}{72\rho^2}\right) := \varepsilon_\rho,\\
\bar{\varepsilon}^2 &\geq \frac{8C_rK}{\bar{c}_1+\ldots+\bar{c}_S}\Big(2t+\delta S\big(\sqrt{B}+1\big)\Big),\\
\bar{\varepsilon}(1-\frac{\bar{\varepsilon}^3}{64}) &> \frac{16C_rSK^2\big(\sqrt{B}+1\big)}{\bar{c}_1+\ldots+\bar{c}_S}\exp\left(-\frac{\big(\beta-\frac{3\bar{\varepsilon}^2}{2}\big)^2}{72\bar{\varepsilon}^2}\right). \qquad (28)
\end{aligned}$$

Choosing $t=\tilde{\varepsilon}^2_{\mu,\rho}\frac{\bar{c}_1+\ldots+\bar{c}_S}{32C_rK}$ and $\delta=\tilde{\varepsilon}^2_{\mu,\rho}\frac{\bar{c}_1+\ldots+\bar{c}_S}{16KS(\sqrt{B}+1)}$ we can merge the first three conditions to $\bar{\varepsilon}\geq\tilde{\varepsilon}^2_{\mu,\rho}$, while following the usual argument, Condition (28) is satisfied once

$$\bar{\varepsilon}\leq\frac{\beta}{\frac{9}{4}+9\sqrt{\log\left(\frac{150K^2S\left(\sqrt{B}+1\right)}{\beta C_r(\bar{c}_1+\ldots+\bar{c}_S)}\right)}}.$$

Given that $\bar{\varepsilon}$ differs at most by δ from ε we see that (27) is larger than zero except with probability

$$2\exp\left(-\frac{N\tilde{\varepsilon}^4_{\mu,\rho}(\bar{c}_1+\ldots+\bar{c}_S)^2}{512\varepsilon^2_{\max}C_r^2S^2K^2\big(\sqrt{B}+1\big)^2}+Kd\log\left(\frac{48\varepsilon_{\max}KS\big(\sqrt{B}+1\big)}{\tilde{\varepsilon}^2_{\mu,\rho}(\bar{c}_1+\ldots+\bar{c}_S)}\right)\right),$$

as long as

$$\varepsilon_{\min}:=\tilde{\varepsilon}_{\mu,\rho}+\frac{\tilde{\varepsilon}^2_{\mu,\rho}}{16K}\leq\varepsilon\leq\varepsilon_{\max}\leq\frac{\beta}{\frac{9}{4}+9\sqrt{\log\left(\frac{150K^2S\left(\sqrt{B}+1\right)}{\beta C_r(\bar{c}_1+\ldots+\bar{c}_S)}\right)}}-\frac{N^{-2q}}{K}. \qquad (29)$$

Again the statement follows from choosing $\varepsilon_{\max}=\varepsilon_{\min}$, making sure that the right most inequality in (29) is satisfied and simplifications. ■

References

A. Agarwal, A. Anandkumar, P. Jain, P. Netrapalli, and R. Tandon. Learning sparsely used overcomplete dictionaries via alternating minimization. *COLT 2014 (arXiv:1310.7991)*, 2014a.

A. Agarwal, A. Anandkumar, and P. Netrapalli. Exact recovery of sparsely used overcomplete dictionaries. *COLT 2014 (arXiv:1309.1952)*, 2014b.

M. Aharon, M. Elad, and A.M. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing.*, 54 (11):4311–4322, November 2006.

S. Arora, R. Ge, and A. Moitra. New algorithms for learning incoherent and overcomplete dictionaries. *COLT 2014 (arXiv:1308.6273)*, 2014.

E. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, 2006.

O. Christensen. *An Introduction to Frames and Riesz Bases*. Birkhäuser, 2003.

D.L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4): 1289–1306, 2006.

D.L. Donoho, M. Elad, and V.N. Temlyakov. Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Transactions on Information Theory*, 52(1): 6–18, January 2006.

D.J. Field and B.A. Olshausen. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996.

Q. Geng, H. Wang, and J. Wright. On the local correctness of ℓ^1-minimization for dictionary learning. *arXiv:1101.5672*, 2011.

P. Georgiev, F.J. Theis, and A. Cichocki. Sparse component analysis and blind source separation of underdetermined mixtures. *IEEE Transactions on Neural Networks*, 16(4): 992–996, 2005.

A. Gersho and R.M. Gray. *Vector Quantization and Signal Compression*. Springer, 1992.

R. Gribonval and K. Schnass. Dictionary identifiability - sparse matrix-factorisation via l_1-minimisation. *IEEE Transactions on Information Theory*, 56(7):3523–3539, July 2010.

R. Gribonval, R. Jenatton, F. Bach, M. Kleinsteuber, and M. Seibert. Sample complexity of dictionary learning and other matrix factorizations. *arXiv:1312.3790*, 2013.

D. Hsu, S.M. Kakade, and T. Zhang. A tail inequality for quadratic forms of subgaussian random vectors. *Electronic Communications in Probability (arXiv:1110.2842)*, 17(14), 2012.

R. Jenatton, F. Bach, and R. Gribonval. Sparse and spurious: dictionary learning with noise and outliers. *arXiv:1407.5155*, 2014.

K. Kreutz-Delgado and B.D. Rao. FOCUSS-based dictionary learning algorithms. In *SPIE 4119*, 2000.

K. Kreutz-Delgado, J.F. Murray, B.D. Rao, K. Engan, T. Lee, and T.J. Sejnowski. Dictionary learning algorithms for sparse representation. *Neural Computations*, 15(2):349–396, 2003.

J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11:19–60, 2010.

A. Maurer and M. Pontil. K-dimensional coding schemes in Hilbert spaces. *IEEE Transactions on Information Theory*, 56(11):5839–5846, 2010.

N.A. Mehta and A.G. Gray. On the sample complexity of predictive sparse coding. *arXiv:1202.4050*, 2012.

M.D. Plumbley. Dictionary learning for ℓ_1-exact sparse coding. In M.E. Davies, C.J. James, and S.A. Abdallah, editors, *International Conference on Independent Component Analysis and Signal Separation*, volume 4666, pages 406–413. Springer, 2007.

R. Rubinstein, A. Bruckstein, and M. Elad. Dictionaries for sparse representation modeling. *Proceedings of the IEEE*, 98(6):1045–1057, 2010.

K. Schnass. On the identifiability of overcomplete dictionaries via the minimisation principle underlying K-SVD. *Applied and Computational Harmonic Analysis*, 37(3):464–491, 2014.

K. Schnass and P. Vandergheynst. Average performance analysis for thresholding. *IEEE Signal Processing Letters*, 14(11):828–831, 2007.

K. Schnass and P. Vandergheynst. Dictionary preconditioning for greedy algorithms. *IEEE Transactions on Signal Processing*, 56(5):1994–2002, 2008.

K. Skretting and K. Engan. Recursive least squares dictionary learning algorithm. *IEEE Transactions on Signal Processing*, 58(4):2121–2130, April 2010.

D. Spielman, H. Wang, and J. Wright. Exact recovery of sparsely-used dictionaries. In *COLT 2012 (arXiv:1206.5882)*, 2012.

J.A. Tropp. On the conditioning of random subdictionaries. *Applied and Computational Harmonic Analysis*, 25(1-24), 2008.

D. Vainsencher, S. Mannor, and A.M. Bruckstein. The sample complexity of dictionary learning. *Journal of Machine Learning Research*, 12(3259-3281), 2011.

R. Vershynin. Introduction to the non-asymptotic analysis of random matrices. In Y. Eldar and G. Kutyniok, editors, *Compressed Sensing, Theory and Applications*, chapter 5. Cambridge University Press, 2012.

M. Yaghoobi, T. Blumensath, and M.E. Davies. Dictionary learning for sparse approximations with the majorization method. *IEEE Transactions on Signal Processing*, 57(6): 2178–2191, June 2009.

M. Zibulevsky and B.A. Pearlmutter. Blind source separation by sparse decomposition in a signal dictionary. *Neural Computations*, 13(4):863–882, 2001.

Journal of Machine Learning Research 16 (2015) 1243-1247 Submitted 11/14; Published 6/15

Encog: Library of Interchangeable Machine Learning Models for Java and C#

Jeff Heaton JEFFHEATON@ACM.ORG
College of Engineering and Computing
Nova Southeastern University
Fort Lauderdale, FL 33314, USA

Editor: Cheng Soon Ong

Abstract

This paper introduces the Encog library for Java and C#, a scalable, adaptable, multi-platform machine learning framework that was first released in 2008. Encog allows a variety of machine learning models to be applied to data sets using regression, classification, and clustering. Various supported machine learning models can be used interchangeably with minimal recoding. Encog uses efficient multithreaded code to reduce training time by exploiting modern multicore processors. The current version of Encog can be downloaded from `http://www.encog.org`.

Keywords: Java, C#, neural network, support vector machine, open source software

1. Intention and Goals

This paper describes the Encog API for Java and C# that is provided as a JAR or DLL library. The C# version of Encog is also compatible with the Xamarin Mono package. Encog has an active community that has provided many enhancements that are beyond the scope of this paper. This includes extensions such as Javascript, GPU processing, C/C++ support, Scala support, and interfaces to various automated trading platforms. The scope of this paper is limited to the Java and C# API.

Encog allows the Java or C# programmer to experiment with a wide range of machine language models using a simple, consistent interface for clustering, regression, and classifications. This allows the programmer to construct applications that discover which model provides the most suitable fit for the data. Encog provides basic tools for automated model selection. Most Encog models are implemented as efficient multithreaded algorithms to reduce processing time. This often allows Encog to perform more efficiently than many other Java and C# libraries, as demonstrated empirically by Taheri (2014) and Matviykiv and Faitas (2012). Luhasz et al. (2013) and Ramos-Pollán et al. (2012) also saw favorable results when evaluating Encog to similar libraries.

The Encog's API is presented in an intuitive object-oriented paradigm that allows various models, optimization algorithms, and training algorithms to be highly interchangeable. However, beneath the API, the models are represented as one and two-dimensional arrays. This internal representation allows for highly efficient calculation. The API shields the programmer from the complexity of model calculation and fitting.

Encog contains nearly 400 unit tests to ensure consistency between the Java and C# model implementations. Expected results are calculated and cross-checked between the two platforms. A custom pseudorandom number generator (PRNG) is used in both language's unit tests to ensure that even stochastic models produce consistent, verifiable test results.

Encog contains nearly 150 examples to demonstrate the use of the API in a variety of scenarios. These examples include simple prediction, time series, simulation, financial applications, path finding, curve fitting, and other applications. Documentation for Encog is provided as Java/C# docs and an online wiki. Additionally, discussion groups and a Stack Overflow tag are maintained for support. Links to all of these resources can be found at `http://www.encog.org`.

2. Framework Overview

The design goal of Encog is to provide interchangeable models with efficient, internal implementations. The Encog framework supports machine learning models with multiple training algorithms. These models are listed here:

- Adaline, Feedforward, Hopfield, PNN/GRNN, RBF & NEAT neural networks
- generalized linear regression (GLM)
- genetic programming (tree-based)
- k-means clustering
- k-nearest neighbors
- linear regression
- self-organizing map (SOM)
- simple recurrent network (Elman and Jordan)
- support vector machine (SVM)

Encog provides optimization algorithms such as particle swarm optimization (PSO) (Poli, 2008), genetic algorithms (GA), Nelder-Mead and simulated annealing. These algorithms can optimize a vector to minimize a loss function; consequently, these algorithms can fit model parameters to data sets.

Propagation-training algorithms for neural network fitting, such as back propagation (Rumelhart et al., 1988), resilient propagation (Riedmiller and Braun, 1992), Levenberg-Marquardt (Marquardt, 1963), quickpropagation (Fahlman, 1988), and scaled conjugate gradient (Møller, 1993) are included. Neural network pruning and model selection can be used to find optimal network architectures. Neural network architectures can be automatically built by a genetic algorithm using NEAT and HyperNEAT (Stanley and Miikkulainen, 2002).

A number of preprocessing tools are built into the Encog library. Collected data can be divided into training, test, and validation sets. Time-series data can be encoded into data windows. Quantitative data can be normalized by range or z-score to prevent biases in some models. Masters (1993) normalizes qualitative data using one-of-n encoding or equilateral encoding. Encog uses these normalization techniques.

Encog also contains extensive support for genetic programming using a tree representation (Koza, 1993). A full set of mathematical and programming functions are provided. Additionally, new functions can be defined. Constant nodes can either be drawn from a constant pool or generated as needed. Rules can optionally be added to simplify expressions and penalize specific genome patterns.

3. API Overview

One of the central design philosophies of Encog is to allow models to be quickly interchanged without a great deal of code modification. A classification example will demonstrate this interchangeability, using the iris data set (Fisher, 1936). Portions of this classification example are presented in this paper, using the Java programming language. The complete example, in both Java and C#, is provided in the *Encog Quick Start Guide* (available from `http://www.encog.org`). The Quick Start Guide also provides regression and time-series examples.

The following example learns to predict the species of an iris flower by using four types of measurements from each flower. To begin, the program loads the iris data set's CSV file. In addition to CSV, Encog contains classes to read fixed-length text, JDBC, ODBC, and XML data sources. The iris data set is loaded, and the four measurement columns are defined as continuous values.

```
VersatileDataSource source = new CSVDataSource(irisFile, false,
    CSVFormat.DECIMAL_POINT);
VersatileMLDataSet data = new VersatileMLDataSet(source);
data.defineSourceColumn("sepal-length", 0, ColumnType.continuous);
data.defineSourceColumn("sepal-width", 1, ColumnType.continuous);
data.defineSourceColumn("petal-length", 2, ColumnType.continuous);
data.defineSourceColumn("petal-width", 3, ColumnType.continuous);
```

The species of Iris is defined as nominal value. Defining the columns as continuous, nominal or ordinal allows Encog to determine the appropriate way to encode these data for a model. For specialized cases, it is possible to override Encog's encoding defaults for any model type.

```
ColumnDefinition outputColumn = data.defineSourceColumn("species", 4,
    ColumnType.nominal);
```

Once the columns have been defined, the file is analyzed to determine minimum, maximum, and other statistical properties of the columns. This allows the columns to be properly normalized and encoded by Encog for modeling.

```
data.analyze();
data.defineSingleOutputOthersInput(outputColumn);
```

Next the model type is defined to be a feedforward neural network.

```
EncogModel model = new EncogModel(data);
model.selectMethod(data, MLMethodFactory.TYPE_FEEDFORWARD);
```

Only the above line needs to be changed to switch to model types that include the following:

- **MLMethodFactory.SVM**: support vector machine
- **MLMethodFactory.TYPE_RBFNETWORK**: RBF neural network

- **MLMethodFactor.TYPE_NEAT**: NEAT neural network
- **MLMethodFactor.TYPE_PNN**: probabilistic neural network

Next the data set is normalized and encoded. Encog will automatically determine the correct normalization type based on the model chosen in the last step. For model validation, 30% of the data are held back. Though the validation sampling is random, a seed of 1001 is used so that the items selected for validation remain constant between program runs. Finally, the default training type is selected.

```
data.normalize();
model.holdBackValidation(0.3, true, 1001);
model.selectTrainingType(data);
```

The example trains using a 5-fold cross-validated technique that chooses the model with the best validation score. The resulting training and validation errors are displayed.

```
MLRegression bestMethod = (MLRegression)model.crossvalidate(5, true);
System.out.println( "Training error: " + EncogUtility.calculateRegressionError(
    bestMethod, model.getTrainingDataset()));
System.out.println( "Validation error: " + EncogUtility.calculateRegressionError
    (bestMethod, model.getValidationDataset()));
```

Display normalization parameters and final model.

```
NormalizationHelper helper = data.getNormHelper();
System.out.println(helper.toString());
System.out.println("Final model: " + bestMethod);
```

4. Future Plans and Conclusions

A number of enhancements are planned for Encog. Gradient boosting machines (GBM) and deep learning are two future model additions. Several planned enhancements will provide interoperability with other machine learning packages. Future versions of Encog will have the ability to read and write Weka Attribute-Relation File Format (ARFF) and libsvm data files. Encog will gain the ability to load and save models in the Predictive Model Markup Language (PMML) format. A code contribution by Mosca (2012) will soon be integrated, enhancing Encog's ensemble learning capabilities.

Acknowledgments

The Encog community has been very helpful for bug reports, bug fixes, and feature suggestions. Contributors to Encog include Olivier Guiglionda, Seema Singh, César Roberto de Souza, and others. A complete list of contributors to Encog can be found at the GitHub repository: `https://github.com/encog`. Alan Mosca, Department of Computer Science and Information Systems, Birkbeck, University of London, UK, created Encog's ensemble functionality. Matthew Dean, Marc Fletcher and Edmund Owen, Semiconductor Physics Research Group, University of Cambridge, UK, created Encog's RBF Neural network model.

References

S. Fahlman. An empirical study of learning speed in back-propagation networks. Technical report, Carnegie Mellon University, 1988.

R. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(7):179–188, 1936.

J. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Complex adaptive systems. MIT Press, 1993. ISBN 978-0-262-11170-6.

G. Luhasz, V. Munteanu, and V. Negru. Data mining considerations for knowledge acquisition in real time strategy games. In *IEEE 11th International Symposium on Intelligent Systems and Informatics, SISY 2013, Subotica, Serbia, September 26-28, 2013*, pages 331–336, 2013. doi: 10.1109/SISY.2013.6662596.

D. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, 11(2):431–441, 1963. doi: 10.1137/0111030.

T. Masters. *Practical Neural Network Recipes in C++*. Academic Press Professional, Inc., San Diego, CA, USA, 1993. ISBN 0-12-479040-2.

O. Matviykiv and O. Faitas. Data classification of spectrum analysis using neural network. *Lviv Polytechnic National University*, 2012.

M. Møller. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, 6(4):525–533, 1993.

A. Mosca. Extending Encog: a study on classifier ensemble techniques. Master's thesis, Birkbeck, University of London, 2012.

R. Poli. Analysis of the publications on the applications of particle swarm optimisation. *J. Artif. Evol. App.*, 2008:4:1–4:10, January 2008. ISSN 1687-6229.

Raúl Ramos-Pollán, Miguel Ángel Guevara-López, and Eugénio C. Oliveira. A software framework for building biomedical machine learning classifiers through grid computing resources. *J. Medical Systems*, 36(4):2245–2257, 2012. doi: 10.1007/s10916-011-9692-3.

M. Riedmiller and H. Braun. Rprop: A fast adaptive learning algorithm. Technical report, Proc. of ISCIS VII), Universitat, 1992.

D. Rumelhart, G. Hinton, and R. Williams. Neurocomputing: Foundations of research. In James A. Anderson and Edward Rosenfeld, editors, *Neurocomputing: Foundations of Research*, chapter Learning Representations by Back-propagating Errors, pages 696–699. MIT Press, Cambridge, MA, USA, 1988. ISBN 0-262-01097-6.

K. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. *Evol. Comput.*, 10(2):99–127, June 2002. ISSN 1063-6560.

T. Taheri. Benchmarking and comparing Encog, Neuroph and JOONE neural networks. `http://goo.gl/A56iyx`, June 2014. Accessed: 2014-10-9.

Journal of Machine Learning Research 16 (2015) 1249-1274 Submitted 4/13; Revised 9/14; Published 7/15

Perturbed Message Passing for Constraint Satisfaction Problems

Siamak Ravanbakhsh MRAVANBA@UALBERTA.CA
Russell Greiner RGREINER@UALBERTA.CA
Department of Computing Science
University of Alberta
Edmonton, AB T6E 2E8, CA

Editor: Alexander Ihler

Abstract

We introduce an efficient message passing scheme for solving Constraint Satisfaction Problems (CSPs), which uses stochastic perturbation of Belief Propagation (BP) and Survey Propagation (SP) messages to bypass decimation and directly produce a single satisfying assignment. Our first CSP solver, called *Perturbed Belief Propagation*, smoothly interpolates two well-known inference procedures; it starts as BP and ends as a Gibbs sampler, which produces a single sample from the set of solutions. Moreover we apply a similar perturbation scheme to SP to produce another CSP solver, *Perturbed Survey Propagation*. Experimental results on random and real-world CSPs show that Perturbed BP is often more successful and at the same time tens to hundreds of times more efficient than standard BP guided decimation. Perturbed BP also compares favorably with state-of-the-art SP-guided decimation, which has a computational complexity that generally scales exponentially worse than our method (w.r.t. the cardinality of variable domains and constraints). Furthermore, our experiments with random satisfiability and coloring problems demonstrate that Perturbed SP can outperform SP-guided decimation, making it the best incomplete random CSP-solver in difficult regimes.

Keywords: constraint satisfaction problem, message passing, belief propagation, survey propagation, Gibbs sampling, decimation

1. Introduction

Probabilistic Graphical Models (PGMs) provide a common ground for recent convergence of themes in computer science (artificial neural networks), statistical physics of disordered systems (spin-glasses) and information theory (error correcting codes). In particular, message passing methods have been successfully applied to obtain state-of-the-art solvers for Constraint Satisfaction Problems (Mézard et al., 2002)

The PGM formulation of a CSP defines a uniform distribution over the set of solutions, where each unsatisfying assignment has a zero probability. In this framework, solving a CSP amounts to producing a sample from this distribution. To this end, usually an inference procedure estimates the marginal probabilities, which suggests an assignment to a subset of the most biased variables. This process of sequentially fixing a subset of variables, called *decimation*, is repeated until all variables are fixed to produce a solution. Due to inaccuracy of the marginal estimates, this procedure gives an incomplete solver (Kautz et al., 2009), in

the sense that the procedure's failure is not a certificate of unsatisfiability. An alternative approach is to use message passing to guide a search procedure that can back-track if a dead-end is reached (*e.g.*, Kask et al., 2004; Parisi, 2003). Here using a branch and bound technique and relying on exact solvers, one may also determine when a CSP is unsatisfiable.

The most common inference procedure for this purpose is Belief Propagation (Pearl, 1988). However, due to geometric properties of the set of solutions (Krzakala et al., 2007) as well as the complications from the decimation procedure (Coja-Oghlan, 2011; Kroc et al., 2009), BP-guided decimation fails on difficult instances. The study of the change in the geometry of solutions has lead to Survey Propagation (Braunstein et al., 2002) which is a powerful message passing procedure that is slower than BP (per iteration) but typically remains convergent, even in many situations when BP fails to converge.

Using decimation, or other search schemes that are guided by message passing, usually requires estimating marginals or partition functions, which is harder than producing a single solution (Valiant, 1979). This paper introduces a message passing scheme to eliminate this requirement, therefore also avoiding the complications of applying decimation. Our alternative has advantage over both BP- and SP-guided decimation when applied to solve CSPs. Here we consider BP and Gibbs Sampling (GS) updates as operators—Φ and Ψ respectively—on a set of messages. We then consider inference procedures that are convex combination (*i.e.*, $\gamma\Psi + (1-\gamma)\Phi$) of these two operators. Our CSP solver, Perturbed BP, starts at $\gamma = 0$ and ends at $\gamma = 1$, smoothly changing from BP to GS, and finally producing a sample from the set of solutions. This change amounts to stochastic biasing the BP messages towards the current estimate of marginals, where this random bias increases in each iteration. This procedure is often much more efficient than BP-guided decimation (BP-dec) and sometimes succeeds where BP-dec fails. Our results on random CSPs (rCSPs) show that Perturbed BP is competitive with SP-guided decimation (SP-dec) in solving difficult random instances.

Since SP can be interpreted as BP applied to an "auxiliary" PGM (Braunstein et al., 2005), we can apply the same perturbation scheme to SP, which we call Perturbed SP. Note that this system, also, does not perform decimation and directly produce a solution (without using local search). Our experiments show that Perturbed SP is often more successful than both SP-dec and Perturbed BP in finding satisfying assignments.

Stochastic variations of BP have been previously proposed to perform inference in graphical models (*e.g.*, Ihler and Mcallester, 2009; Noorshams and Wainwright, 2013). However, to our knowledge, Perturbed BP is the first method to directly combine GS and BP updates.

In the following, Section 1.1 introduces PGM formulation of CSP using factor-graph notation. Section 1.2 reviews the BP equations and decimation procedure, then Section 1.3 casts GS as a message update procedure. Section 2 introduces Perturbed BP as a combination of GS and BP. Section 2.1 compares BP-dec and Perturbed BP on benchmark CSP instances, showing that our method is often several folds faster and more successful in solving CSPs. Section 3 overviews the geometric properties of the set of solutions of rCSPs, then reviews first order Replica Symmetry Breaking Postulate and the resulting SP equations for CSP. Section 3.2 introduces Perturbed SP and Section 3.3 presents our experimental results for random satisfiability and random coloring instances close to the unsatisfiability threshold. Finally, Section 3.4 further discusses the behavior of decimation and perturbed BP in the light of a geometric picture of the set of solutions and the experimental results.

1.1 Factor Graph Representation of CSP

Let $x = (x_1, x_2, \ldots, x_N)$ be a tuple of N discrete variables $x_i \in \mathcal{X}_i$, where each $\mathcal{X}_i$ is the domain of x_i. Let $I \subseteq \mathcal{N} = \{1, 2, \ldots, N\}$ denote a subset of variable indices and $x_I = \{x_i \,|\, i \in I\}$ be the (sub)tuple of variables in x indexed by the subset I. Each constraint $C_I(x_I) : \left(\prod_{i \in I} \mathcal{X}_i\right) \to \{0, 1\}$ maps an assignment to 1 *iff* that assignment satisfies that constraint. Then the normalized product of all constraints defines a uniform distribution over solutions

$$\mu(x) \quad \triangleq \quad \frac{1}{Z} \prod_I C_I(x_I) \tag{1}$$

where the partition function $Z = \sum_{\mathcal{X}} \prod_I C_I(x_I)$ is equal to the number of solutions.[1] Notice that $\mu(x)$ is non-zero *iff* all of the constraints are satisfied—that is x is a solution. With slight abuse of notation we will use probability density and probability distribution interchangeably.

Example 1 (q-COL:) *Here, $x_i \in \mathcal{X}_i = \{1, \ldots, q\}$ is a q-ary variable for each $i \in \mathcal{N}$, and we have M constraints; each constraint $C_{i,j}(x_i, x_j) = 1 - \delta(x_i, x_j)$ depends only on two variables and is satisfied iff the two variables have different values (colors). Here $\delta(x, x')$ is equal to 1 if $x = x'$ and 0 otherwise.*

This model can be conveniently represented as a bipartite graph, known as a *factor graph* (Kschischang et al., 2001), which includes two sets of nodes: variable nodes x_i, and constraint (or factor) nodes C_I. A variable node i (note that we will often identify a variable "x_i" with its index "i") is connected to a constraint node I if and only if $i \in I$. We will use ∂ to denote the neighbors of a variable or constraint node in the factor graph—that is $\partial I = \{i \;\mid\; i \in I\}$ (which is the set I) and $\partial i = \{I \;\mid\; i \in I\}$. Finally we use Δi to denote the Markov blanket of node x_i ($\Delta i = \{j \in \partial I \;\mid\; I \in \partial i, j \neq i\}$).

The marginal of $\mu(\cdot)$ for variable x_i is defined as

$$\mu(x_i) \;\triangleq\; \sum_{\mathcal{X}_{\mathcal{N} \setminus i}} \mu(x)$$

where the summation above is over all variables but x_i. Below, we use $\widehat{\mu}(x_i)$ to denote an *estimate* of this marginal. Finally, we use $\mathcal{S}$ to denote the (possibly empty) set of solutions $\mathcal{S} = \{x \in \mathcal{X} \;\mid\; \mu(x) \neq 0\}$.

Example 2 (κ-SAT:) *All variables are binary ($\mathcal{X}_i = \{True, False\}$) and each clause (constraint C_I) depends on $\kappa = |\partial I|$ variables. A clause evaluates to 0 only for a single assignment out of 2^κ possible assignment of variables (Garey and Johnson, 1979).*

Consider the following 3-SAT problem over 3 variables with 5 clauses

$$\begin{aligned} SAT(x) = &\underbrace{(\neg x_1 \vee \neg x_2 \vee x_3)}_{C_1} \wedge \underbrace{(\neg x_1 \vee x_2 \vee x_3)}_{C_2} \wedge \underbrace{(x_1 \vee \neg x_2 \vee x_3)}_{C_3} \\ &\wedge \underbrace{(\neg x_1 \vee x_2 \vee \neg x_3)}_{C_4} \wedge \underbrace{(x_1 \vee \neg x_2 \vee \neg x_3)}_{C_5}. \end{aligned} \tag{2}$$

1. For Equation (1) to remain valid when the CSP is unsatisfiable, we define $\frac{0}{0} \triangleq 0$.

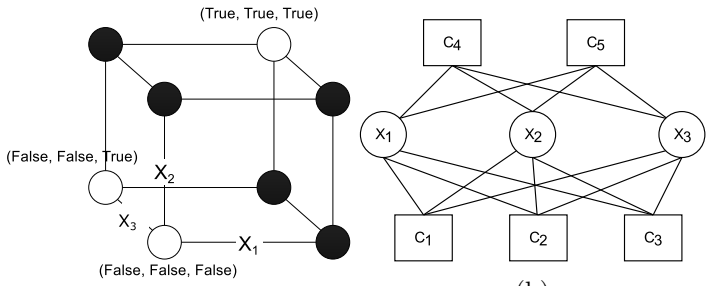


Figure 1: (a) The set of all possible assignments to 3 variables. The solutions to the 3-SAT problem of Equation (2) are in white circles. (b) The factor-graph corresponding to the 3-SAT problem of Equation (2). Here each factor prohibits a single assignment.

The constraint corresponding to the first clause takes the value 1*, except for* $x = \{True, True, False\}$*, in which case it is equal to* 0*. The set of solutions for this problem is given by* $\mathcal{S} = \big\{(True, True, True),\ (False, False, False),\ (False, False, True)\big\}$*. Figure 1 shows the solutions as well as the corresponding factor graph.*[2]

1.2 Belief Propagation-guided Decimation

Belief Propagation (Pearl, 1988) is a recursive update procedure that sends a sequence of messages from variables to constraints ($\nu_{i\to I}$) and vice-versa ($\nu_{I\to i}$)

$$\nu_{i\to I}(x_i) \ \propto \prod_{J\in\partial i\setminus I} \nu_{J\to i}(x_i) \tag{3}$$

$$\nu_{I\to i}(x_i) \ \propto \sum_{x_{I\setminus i}\in\mathcal{X}_{\partial I\setminus i}} C_I(x_I) \prod_{j\in\partial I\setminus i} \nu_{j\to I}(x_j) \tag{4}$$

where $J \in \partial i \setminus I$ refers to all the factors connected to variable x_i, except for factor C_I. Similarly the summation in Equation (4) is over $\mathcal{X}_{\partial I\setminus i}$, means we are summing out all x_j that are connected to C_I (*i.e.*, x_j *s.t.* $j \in I \setminus i$) except for x_i.

The messages are typically initialized to a uniform or a random distribution. This recursive update of messages is usually performed until convergence—*i.e.*, until the maximum change in the value of all messages, from one iteration to the next, is negligible (*i.e.*, below some small ϵ). At any point during the updates, the estimated marginal probabilities are given by

$$\widehat{\mu}(x_i) \ \propto \prod_{J\in\partial i} \nu_{J\to i}(x_i). \tag{5}$$

2. In this simple case, we could combine all the constraints into a single constraint over 3 variables and simplify the factor graph. However, in general SAT, this cost-saving simplification is often not possible.

In a factor graph without loops, each BP message summarizes the effect of the (sub-tree that resides on the) sender-side on the receiving side.

Example 3 *Applying BP to the 3-SAT problem of Equation* (2) *takes 20 iterations to converge (i.e., for the maximum change in the marginals to be below* $\epsilon = 10^{-9}$*). Here the message,* $\nu_{C_1 \to 1}(x_1)$*, from* $C1$ *to* x_1 *is*

$$\nu_{C_1 \to 1}(x_1) \quad \propto \quad \sum_{x_{2,3}} C_1(x_{1,2,3})\, \nu_{2 \to C_1}(x_2)\, \nu_{3 \to C_1}(x_3)$$

and similarly, the message in the opposite direction, $\nu_{1 \to C_1}(x_1)$*, is defined as*

$$\nu_{1 \to C_1}(x_1) \quad \propto \quad \nu_{C_2 \to 1}(x_1)\, \nu_{C_3 \to 1}(x_1)\, \nu_{C_4 \to 1}(x_1)\, \nu_{C_5 \to 1}(x_1).$$

Here BP gives us the following approximate marginals: $\widehat{\mu}(x_1 = True) = \widehat{\mu}(x_2 = True) = .319$ *and* $\widehat{\mu}(x_3 = True) = .522$*. From the set of solutions, we know that the correct marginals are* $\widehat{\mu}(x_1 = True) = \widehat{\mu}(x_2 = True) = 1/3$ *and* $\widehat{\mu}(x_3 = True) = 2/3$*. The error of BP is caused by influential loops in the factor-graph of Figure 1(b). Here the error is rather small; it can be arbitrarily large in some instances or BP may not converge at all.*

The time complexity of BP updates of Equation (3) and Equation (4), for each of the messages exchanged between i and I, are $\mathcal{O}(|\partial i|\ |\mathcal{X}_i|)$ and $\mathcal{O}(|\mathcal{X}_I|)$ respectively. We may reduce the time complexity of BP by synchronously updating all the messages $\nu_{i \to I}\ \forall I \in \partial i$ that leave node i. For this, we first calculate the beliefs $\widehat{\mu}(x_i)$ using Equation (5) and produce each $\nu_{i \to I}$ using

$$\nu_{i \to I}(x_i) \ \propto \ \frac{\widehat{\mu}(x_i)}{\nu_{I \to i}(x_i)}. \tag{6}$$

Note than we can substitute Equation (4) into Equation (3) and Equation (5) and only keep variable-to-factor messages. After this substitution, BP can be viewed as a fixed-point iteration procedure that repeatedly applies the operator $\Phi(\{\nu_{i \to I}\}) \triangleq \{\Phi_{i \to I}(\{\nu_{j \to J}\}_{j \in \Delta i, J \in \partial i \setminus I})\}_{i, I \in \partial i}$ to the set of messages in hope of reaching a fixed point—that is

$$\nu_{i \to I}(x_i) \ \propto \ \prod_{J \in \partial i \setminus I} \sum_{\mathcal{X}_{\partial J \setminus i}} C_J(x_J) \prod_{j \in \partial J \setminus i} \nu_{j \to J}(x_j) \ \triangleq \ \Phi_{i \to I}(\{\nu_{j \to J}\}_{j \in \Delta i, J \in \partial i \setminus I})(x_i) \tag{7}$$

and therefore Equation (5) becomes

$$\widehat{\mu}(x_i) \quad \propto \quad \prod_{I \in \partial i} \sum_{\mathcal{X}_{\partial I \setminus i}} C_I(x_I) \prod_{j \in \partial I \setminus i} \nu_{j \to I}(x_j) \tag{8}$$

where $\Phi_{i \to I}$ denotes individual message update operators. We let operator $\Phi(.)$ denote the set of these $\Phi_{i \to I}$ operators.

1.2.1 Decimation

The decimation procedure can employ BP (or SP) to solve a CSP. We refer to the corresponding method as BP-dec (or SP-dec). After running the inference procedure and obtaining $\widehat{\mu}(x_i)$, $\forall i$, the decimation procedure uses a heuristic approach to select the most biased variables (or just a random subset) and fixes these variables to their most biased values (or a random $\widehat{x}_i \sim \widehat{\mu}(x_i)$). If it selects a fraction ρ of remaining variables to fix after each convergence, this multiplies an additional $\log_{\frac{1}{\rho}}(N)$ to the linear (in N) cost[3] for each iteration of BP (or SP). The following algorithm 1 summarizes BP-dec with a particular scheduling of updates:

input : factor-graph of a CSP
output: a satisfying assignment x^* if an assignment was found. UNSATISFIED otherwise

1 initialize the messages
2 $\widetilde{\mathcal{N}} \leftarrow \mathcal{N}$ (set of all variable indices)
3 **while** $\widetilde{\mathcal{N}}$ *is not empty* **do** `// decimation loop`
4
5 **repeat**`// BP loop`
6
7 **foreach** $i \in \widetilde{\mathcal{N}}$ **do**
8 calculate messages $\{\nu_{I\to i}\}_{I\in\partial i}$ using Equation (4)
9 **if** $\{\nu_{I\to i}\}_{I\in\partial i}$ *are contradictory* **then**
return : UNSATISFIED
10 calculate marginal $\widehat{\mu}(x_i)$ using Equation (5)
11 calculate messages $\{\nu_{i\to I}\}_{I\in\partial i}$ using Equation (3) or Equation (6)
12 **until** *convergence*
13 select $\mathcal{B} \subseteq \widetilde{\mathcal{N}}$ using $\{\widehat{\mu}(x_i)\}_{i\in\widetilde{\mathcal{N}}}$
14 fix $x_j^* \leftarrow \arg_{x_j}\max \widehat{\mu}(x_j) \quad \forall j \in \mathcal{B}$
15 reduce the constraints $\{C_I\}_{I\in\partial j}$ for every $j \in \mathcal{B}$
return : $x^* = (x_1^*, \ldots, x_N^*)$

Algorithm 1: Belief Propagation-guided Decimation (BP-dec)

The condition of line 9 is satisfied iff the product of incoming messages to node i is 0 for all $x_i \in \mathcal{X}_i$. This means that neighboring constraints have strict disagreement about the value of x_i and the decimation has found a contradiction. This contradiction can happen because, either (I) there is no solution for the reduced problem even if the original problem had a solution, or (II) the reduced problem has a solution but the BP messages are inaccurate.

Example 4 *To apply BP-dec to previous example, we first calculate BP marginals, as shown in the example above. Here $\widehat{\mu}(x_1)$ and $\widehat{\mu}(x_2)$ have the highest bias. By fixing the*

3. Assuming the number of edges in the factor graph are in the order of N. In general, using synchronous update of Equation (6) and assuming a constant factor cardinality, $|\partial I|$, the cost of each iteration is $\mathcal{O}(E)$, where E is the number of edges in the factor-graph.

value of x_1 to $False$, the SAT problem of Equation (2) *collapses to*

$$SAT(x_{\{2,3\}})|_{x_1=False} \;=\; (\neg x_2 \vee x_3) \wedge (\neg x_2 \vee \neg x_3).$$

BP-dec applies BP again to this reduced problem, which give $\widehat{\mu}(x_2 = True) = .14$ (note here that $\mu(x_2 = True) = 0$) and $\widehat{\mu}(x_3 = True) = 1/2$. By fixing x_2 to $False$, another round of decimation yields a solution $x^* = (False, False, True)$.

1.3 Gibbs Sampling as Message Update

Gibbs Sampling (GS) is a Markov Chain Monte Carlo (MCMC) inference procedure (Andrieu et al., 2003) that can produce a set of samples $\widehat{x}[1], \ldots, \widehat{x}[L]$ from a given PGM. We can then recover the marginal probabilities, as empirical expectations

$$\widehat{\mu}^L(x_i) \quad \propto \quad \frac{1}{L}\sum_{n=1}^{L} \delta(\widehat{x}[n]_i, x_i). \tag{9}$$

Our algorithm only considers a single particle $\widehat{x} = \widehat{x}[1]$. GS starts from a random initial state $\widehat{x}^{(t=0)}$ and at each time-step t, updates each $\widehat{x}_i$ by sampling from:

$$\widehat{x}_i^{(t)} \;\sim\; \mu(x_i) \quad \propto \quad \prod_{I \in \partial i} C_I(x_i, \widehat{x}_{\partial I \setminus i}^{(t-1)}) \tag{10}$$

If the Markov chain satisfies certain basic properties (Robert and Casella, 2005), $x_i^{(\infty)}$ is guaranteed to be an unbiased sample from $\mu(x_i)$ and therefore our marginal estimate, $\widehat{\mu}^L(x_i)$, becomes exact as $L \to \infty$.

In order to interpolate between BP and GS, we establish a correspondence between a particle in GS and a set of variable-to-factor messages—*i.e.*, $\widehat{x} \Leftrightarrow \{\nu_{i\to I}(.)\}_{i, I\in\partial i}$. Here all the messages leaving variable x_i are equal to a δ-function defined based on $\widehat{x}_i$

$$\nu_{i\to I}(x_i) \;=\; \delta(x_i, \widehat{x}_i) \quad \forall I \in \partial i.$$

We define the random GS operator $\Psi = \{\Psi_i\}_i$ and rewrite the GS update of Equation (10) as

$$\nu_{i\to I}(x_i) \quad \triangleq \quad \Psi_i(\{\nu_{j\to J}(x_j)\}_{j\in\Delta i, J\in\partial i})(x_i) \quad = \quad \delta(\widehat{x}_i, x_i) \tag{11}$$

where $\widehat{x}_i$ is sampled from

$$\begin{aligned} \widehat{x}_i \;\sim\; \widehat{\mu}(x_i) \quad &\propto \quad \prod_{J\in\partial i} C_I(x_i, \widehat{x}_{\partial I\setminus i}) \\ &\propto \quad \prod_{I\in\partial i} \sum_{\mathcal{X}_{\partial I\setminus i}} C_I(x_I) \prod_{j\in\partial I\setminus i} \nu_{j\to I}(x_j). \end{aligned} \tag{12}$$

Note that Equation (12) is identical to BP estimate of the marginal Equation (8). This equality is a consequence of the way we have defined messages in the GS update and allows us to combine BP and GS updates in the following section.

2. Perturbed Belief Propagation

Here we introduce an alternative to decimation that does not require repeated application of inference. The idea is to use a linear combination of BP and GS operators (Equation 7 and Equation 11) to update the messages

$$\Gamma(\{\nu_{i\to I}\}) \triangleq \gamma\ \Psi(\{\nu_{i\to I}\}) + (1-\gamma)\Phi(\{\nu_{i\to I}\}).$$

The Perturbed BP operator $\Gamma = \{\Gamma_{i\to I}\}_{i,I\in\partial i}$ updates each message by calculating the outgoing message according to BP and GS operators and linearly combines them to get the final message. During T iterations of Perturbed BP, the parameter γ is gradually and linearly changed from 0 towards 1. Algorithm 2 below summarizes this procedure.

input : factor graph of a CSP, number of iterations T
output: a satisfying assignment x^* if an assignment was found. UNSATISFIED otherwise

1 initialize the messages
2 $\gamma \leftarrow 0$
3 $\widetilde{\mathcal{N}} \leftarrow \mathcal{N}$ (set of all variable indices)
4 **for** $t = 1$ *to* T **do**
5 **foreach** *variable* x_i **do**
6 calculate $\nu_{I\to i}$ using Equation (4) $\forall I \in \partial i$
7 **if** $\{\nu_{I\to i}\}_{I\in\partial i}$ *are contradictory* **then**
 return : UNSATISFIED
8 calculate marginals $\widehat{\mu}(x_i)$ using Equation (12)
9 calculate BP messages $\nu_{i\to I}$ using Equation (3) or Equation (6) $\forall I \in \partial i$.
10 sample $\widehat{x}_i \sim \widehat{\mu}(x_i)$
11 combine BP and Gibbs sampling messages:
 $\nu_{i\to I} \leftarrow \gamma\ \nu_{i\to I} + (1-\gamma)\ \delta(x_i, \widehat{x}_i)$
12 $\gamma \leftarrow \gamma + \frac{1}{T-1}$
return : $x^* = \{x_1^*, \ldots, x_N^*\}$

Algorithm 2: Perturbed Belief Propagation

In step 7, if the product of incoming messages is 0 for all $x_i \in \mathcal{X}_i$ for some i, different neighboring constraints have strict disagreement about x_i; therefore this run of Perturbed BP will not be able to satisfy this CSP. Since the procedure is inherently stochastic, if the CSP is satisfiable, re-application of the same procedure to the problem may avoid this specific contradiction.

2.1 Experimental Results on Benchmark CSP

This section compares the performance of BP-dec and Perturbed BP on benchmark CSPs. We considered CSP instances from XCSP repository (Roussel and Lecoutre, 2009; Lecoutre, 2013), without global constraints or complex domains.[4]

4. All instances with intensive constraints (*i.e.*, functional form) were converted into extensive format for explicit representation using dense factors. We further removed instances containing constraints with

We used a convergence threshold of $\epsilon = .001$ for BP and terminated if the threshold was not reached after $T = 10 \times 2^{10} = 10,240$ iterations. To perform decimation, we sort the variables according to their bias and fix ρ fraction of the most biased variables in each iteration of decimation. This fraction, ρ, was initially set to 100%, and it was divided by 2 each time BP-dec failed on the same instance. BP-dec was repeatedly applied using the reduced ρ, at most 10 times, unless a solution was reached (that is $\rho = .1\%$ at final attempt).

For Perturbed BP, we set $T = 10$ at the starting attempt, which was increased by a factor of 2 in case of failure. This was repeated at most 10 times, which means Perturbed BP used $T = 10,240$ at its final attempt. Note that Perturbed BP at most uses the same number of iterations as the maximum iterations per single iteration of decimation in BP-dec.

Figure 2(a,b) compares the time and iterations of BP-dec and Perturbed BP for successful attempts where both methods satisfied an instance. The result for individual problem-sets is reported in the appendix.

Empirically, we found that Perturbed BP both solved (slightly) more instances than BP-dec (284 vs 253), and was (hundreds of times) more efficient: while Perturbed BP required only 133 iterations on average, BP-dec required an average of 41,284 iterations for successful instances.

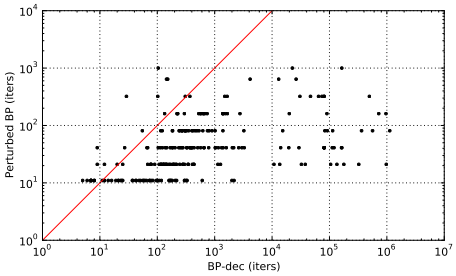
We also ran BP-dec on all the benchmarks with maximum number of iterations set to $T = 1000$ and $T = 100$ iterations. This reduced the number of satisfied instances to 249 for $T = 1000$ and 247 for $T = 100$, but also reduced the average number of iterations to 1570 and 562 respectively, which are still several folds more expensive than Perturbed BP. Figure 2(c-f) compare the time and iterations used by BP-dec in these settings with that of Perturbed BP, when both methods found a satisfying assignment. See the appendix for a more detailed report on these results.

3. Critical Phenomena in Random CSPs

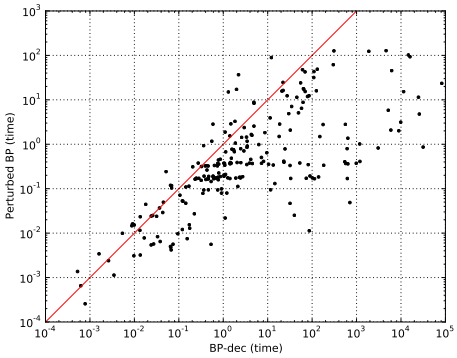
Random CSP (rCSP) instances have been extensively used in order to study the properties of combinatorial problems (Mitchell et al., 1992; Achioptas and Sorkin, 2000; Krzakala et al., 2007) as well as in analysis and design of algorithms (*e.g.*, Selman et al., 1994; Mézard et al., 2002). Random CSPs are closely related to spin-glasses in statistical physics (Kirkpatrick and Selman, 1994; Fu and Anderson, 1986). This connection follows from the fact that the Hamiltonian of these spin-glass systems resembles the objective functions in many combinatorial problems, which decompose to pairwise (or higher order) interactions, allowing for a graphical representation in the form of a PGM. Here message passing methods, such as belief propagation (BP) and survey propagation (SP), provide consistency conditions on locally tree-like neighborhoods of the graph.

The analogy between a physical system and computational problem extends to their critical behavior where computation relates to dynamics (Ricci-Tersenghi, 2010). In com-

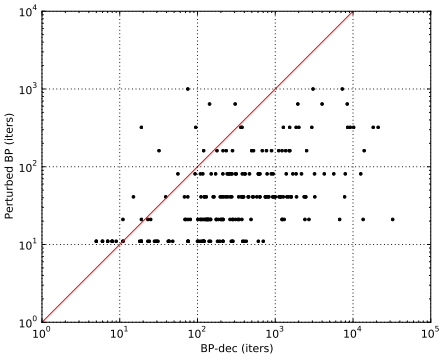
more than 10^6 entries in their tabular form. We also discarded instances that collectively had more than 10^8 entries in the dense tabular form of their constraints. Since our implementation represents all factors in a dense tabular form, we had to remove many instances because of their large factor size. We anticipate that Perturbed BP and BP-dec could probably solve many of these instances using a sparse representation.



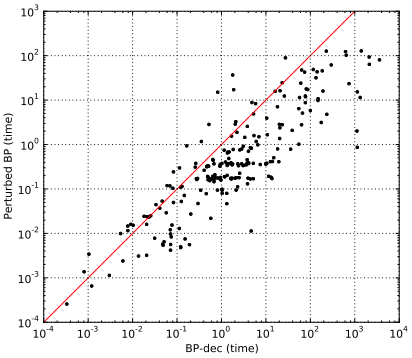
(a) iterations



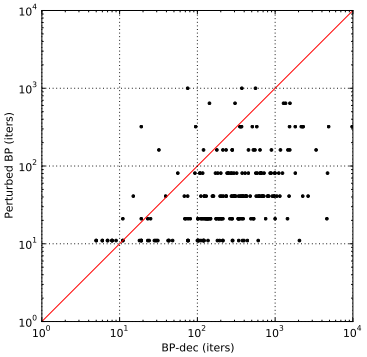
(b) time (seconds)



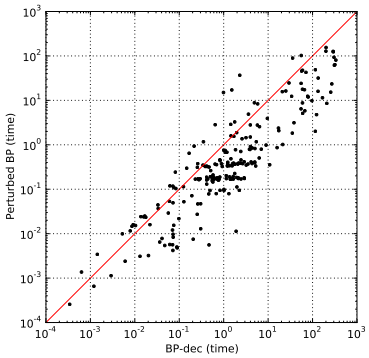
(c) iterations for $T = 1000$



(d) time (seconds) for $T = 1000$



(e) iterations for $T = 100$



(f) time (seconds) for $T = 100$

Figure 2: Comparison of time and number of iterations used by BP-dec and Perturbed BP in benchmark instances where both methods found satisfying assignments. (a,b) Maximum number of BP iterations per iteration of decimation is $T = 10240$, equal to the maximum iterations used by Perturbed BP. (c,d) Maximum number of iterations for BP in BP-dec is reduced to $T = 1000$. (e,f) Maximum number of iterations for BP in BP-dec is further reduced to $T = 100$.

puter science, this critical behavior is related to the time-complexity of algorithms employed to solve such problems, while in spin-glass theory this translates to dynamics of glassy state, and exponential relaxation times (Mézard et al., 1987). In fact, this connection has been used to attempt to prove the conjecture that $\mathcal{P}$ is not equal to $\mathcal{NP}$ (Deolalikar, 2010).

Studies of rCSP, as a critical phenomena, focus on the geometry of the solution space as a function of the problem's difficulty, where rigorous (*e.g.*, Achlioptas and Coja-Oghlan, 2008; Cocco et al., 2003) and non-rigorous (*e.g.*, cavity method of Mézard and Parisi, 2001, 2003) analyses have confirmed the same geometric picture.

When working with large random instances, a scalar α associated with a problem instance, a.k.a. control parameter—for example, the clause to variable ratio in SAT—can characterize that instance's difficulty (*i.e.*, larger control parameter corresponds to a more difficult instance) and in many situations it characterizes a sharp transition from satisfiability to unsatisfiability (Cheeseman et al., 1991).

Example 5 (Random κ-SAT) *Random κ-SAT instance with N variables and $M = \alpha N$ constraints are generated by selecting κ variables at random for each constraint. Each constraint is set to zero (i.e., unsatisfied) for a single random assignment (out of 2^κ). Here α is the control parameter.*

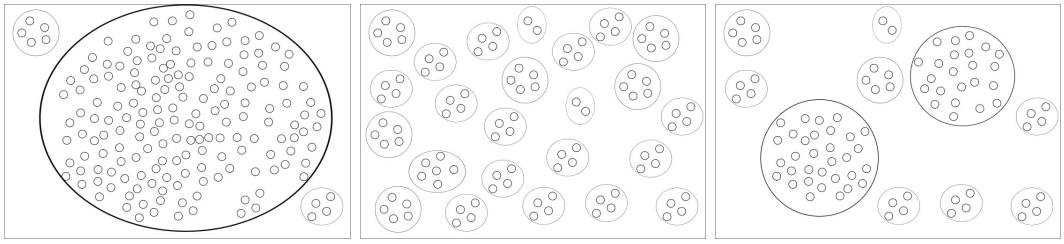
Example 6 (Random q-COL) *The control parameter for a random q-COL instances with N variables and M constraints is its average degree $\alpha = \frac{2M}{N}$. We consider Erdős-Rény random graphs and generate a random instance by sequentially selecting two distinct variables out of N at random to generate each of M edges. For large N, this is equivalent to selecting each possible factor with a fixed probability, which means the nodes have Poisson degree distribution $\mathbb{P}(|\partial i| = d) \propto e^{-\alpha}\alpha^d$.*

While there are tight bounds for some problems (*e.g.*, Achlioptas et al., 2005), finding the exact location of this transition for different CSPs is still an open problem. Besides transition to unsatisfiability, these analyses has revealed several other (phase) transitions (Krzakala et al., 2007). Figure 3(a)-(c) shows how the geometry of the set of solutions changes by increasing the control parameter.

Here we enumerate various phases of the problem for increasing values of the control parameter: **(a)** In the so-called *Replica Symmetric* (RS) phase, the symmetries of the set of solutions (a.k.a. ground states) reflect the trivial symmetries of problem w.r.t. variable domains. For example, for q-COL the set of solutions is symmetric w.r.t. swapping all red and blue assignment. In this regime, the set of solutions form a giant cluster (*i.e.*, a set of neighboring solutions), where two solutions are considered neighbors when their Hamming distance is one (Achlioptas and Coja-Oghlan, 2008) or non-divergent with number of variables (Mézard and Parisi, 2003). Local search methods (*e.g.*, Selman et al., 1994) and BP-dec can often efficiently solve random CSPs that belong to this phase.

(b) In *clustering* or *dynamical* transition (1dRSB[5]), the set of solutions decomposes into an exponential number of distant clusters. Here two clusters are distant if the Hamming distance between their respective members is divergent (*e.g.*, linear) in the number

5. 1dRSB stands for 1st order dynamical RSB. The term Replica Symmetry Breaking (RSB) originates from the technique—*i.e.*, Replica trick (Mézard et al. 1987)—that was first used to analyze this setting. According to RSB, the trivial symmetries of the problem do not characterize the clusters of solution.



(a) Replica Symmetric (b) clustering (c) condensation

Figure 3: A 2-dimensional schematic view of how the set of solutions of CSP varies as we increase the control parameter α from (a) replica symmetric phase to (b) clustering phase to (c) condensation phase. Here small circles represent solutions and the bigger circles represent clusters of solutions. Note that this view is very simplistic in many ways; for example, the total number of solutions and the size of clusters should generally decrease from (a) to (c).

of variables. **(c)** In the *condensation* phase transition (1sRSB[6]), the set of solutions condenses into a few dominant clusters. Dominant clusters have roughly the same number of solutions and they collectively contain almost all of the solutions. While SP can be used even within the condensation phase, BP usually fails to converge in this regime. However each cluster of solutions in the clustering and condensation phase is a valid fixed-point of BP, which is called a "quasi-solution" of BP. **(d)** A *rigidity* transition (not included in Figure 3) identifies a phase in which a finite portion of variables are fixed within dominant clusters. This transition triggers an exponential decrease in the total number of solutions, which leads to **(e)** unsatisfiability transition.[7] This rough picture summarizes first order Replica Symmetry Breaking's (1RSB) basic assumptions (Mézard and Montanari, 2009).

From a geometric perspective, the intuitive idea behind Perturbed BP, is to perturb the messages towards a solution. However, in order to achieve this, we need to initialize the messages to a proper neighborhood of a solution. Since these neighborhoods are not initially known, we resort to stochastic perturbation of messages to make local marginals more biased towards a subspace of solutions. This continuous perturbation of all messages is performed in a way that allows each BP message to re-adjust itself to the other perturbations, more and more focusing on a random subset of solutions.

3.1 1RSB Postulate and Survey Propagation

Large random graphs are locally tree-like, which means the length of short loops are typically in the order of $\log(N)$ (Janson et al., 2001). This ensures that, in the absence of long-range correlations, BP is asymptotically exact, as the set of messages incoming to each node or factor are almost independent. Although BP messages remain uncorrelated until the condensation transition (Krzakala et al., 2007), the BP equations do not completely characterize the set of solutions after the clustering transition. This inadequacy is indicated

6. 1sRSB is short for 1st order static Replica Symmetry Breaking.
7. In some problems, the rigidity transition occurs before condensation transition.

by the existence of a set of several valid fixed points (rather than a unique fixed-point) for BP, each of which corresponds to a quasi-solution. For a better intuition, consider the cartoons of Figures 3(b) and (c). During the clustering phase (b), x_i and x_j (corresponding to the x and y axes) are not highly correlated, but they become correlated during and after condensation (c). This correlation between variables that are far apart in the PGM results in correlation between the BP messages. This violates BP's assumption that messages are uncorrelated, which results in BP's failure in this regime.

1RSB's approach to incorporating this clustering of solutions into the equilibrium conditions is to define a new Gibbs measure over clusters. Let $\mathbf{y} \subset \mathcal{S}$ denote a cluster of solutions and $\mathcal{Y}$ be the set of all such clusters. The idea is to treat $\mathcal{Y}$ the same as we treated $\mathcal{X}$, by defining a distribution

$$\boldsymbol{\mu}(\mathbf{y}) \quad \propto \quad |\mathbf{y}|^m \quad \forall\, \mathbf{y} \in \mathcal{Y} \tag{13}$$

where $m \in [0, 1]$, called the Parisi parameter (Mézard et al., 1987), specifies how each cluster's weight depends on its size. This implicitly defines a distribution over $\mathcal{X}$

$$\mu(x) \quad \propto \quad \sum_{\mathbf{y} \ni x} \boldsymbol{\mu}(\mathbf{y}) \tag{14}$$

N.b., $m = 1$ corresponds to the original distribution (Equation (1)).

Example 7 *Going back to our simple 3-SAT example,* $\mathbf{y}^{(1)} = \{(True, True, True)\}$ *and* $\mathbf{y}^{(2)} = \{(False, False, False),\ (False, False, True)\}$ *are two clusters of solutions. Using* $m = 1$*, we have* $\boldsymbol{\mu}(\{\{True, True, True\}\}) = 1/3$ *and* $\boldsymbol{\mu}(\{\{False, False, False\}, \{False, False, True\}\}) = 2/3$*. This distribution over clusters reproduces the distribution over solutions—i.e.,* $\mu(x) = 1/3\ \forall x \in \mathcal{S}$*. On the other hand, using* $m = 0$*, produces a uniform distribution over clusters, but it does not give us a uniform distribution over the solutions.*

This meta-construction for $\mu(\mathbf{y})$ can be represented using an auxiliary PGM. One may use BP to find marginals over this PGM; here BP messages are distributions over all BP messages in the original PGM, as each cluster is a fixed-point for BP. This requirement to represent a distribution over distributions makes 1RSB practically intractable. In general, each original BP message is a distribution over $\mathcal{X}_i$ and it is difficult to define a distribution over this infinite set. However this simplifies if the original BP messages can have limited values. Fortunately if we apply max-product BP to solve a CSP, instead of sum-product BP (of Equations (3) and (4)), the messages can have a finite set of values.

Max-Product BP: Our previous formulation of CSP was using sum-product BP. In general, max-product BP is used to find the Maximum a Posteriori (MAP) assignment in a PGM, which is a single assignment with the highest probability. In our PGM, the MAP assignment is a solution for the CSP. The max-product update equations are

$$\begin{aligned}
\eta_{i \to I}(x_i) &= \textstyle\prod_{J \in \partial i \setminus I} \eta_{J \to i}(x_i) &&= \Lambda_{i \to I}(\ \{\eta_{J \to i}\}_{J \in \partial i \setminus I}\ )(x_i) && (15)\\
\eta_{I \to i}(x_i) &= \max_{\mathcal{X}_{\partial I \setminus i}} C_I(x_I) \textstyle\prod_{j \in \partial I \setminus i} \eta_{j \to I}(x_j) &&= \Lambda_{I \to i}(\ \{\eta_{j \to I}\}_{j \in \partial I \setminus i}\ )(x_i) && (16)\\
\widehat{\mu}(x_i) &= \textstyle\prod_{J \in \partial i} \eta_{J \to i}(x_i) &&= \Lambda_i(\ \{\eta_{J \to i}\}_{J \in \partial i}\ )(x_i) && (17)
\end{aligned}$$

where $\Lambda = \{\Lambda_{i\to I}, \Lambda_{I\to i}\}_{i,I\in\partial I}$ is the max-product BP operator and Λ_i represents the marginal estimate as a function of messages. Note that here messages and marginals are not distributions. We initialize $\nu_{i\to I}(x_i) \in \{0,1\},\ \forall I, i \in \partial I, x_i \in \mathcal{X}_i$. Because of the way constraints and update equations are defined, at any point during the updates we have $\nu_{i\to I}(x_i) \in \{0,1\}$. This is also true for $\widehat{\mu}(x_i)$. Here any of $\nu_{i\to I}(x_i) = 1$, $\nu_{I\to i}(x_i) = 1$ or $\widehat{\mu}(x_i) = 1$, shows that value x_i is allowed according to a message or marginal, while 0 forbids that value. Note that $\widehat{\mu}(x_i) = 0\ \forall x_i \in \mathcal{X}_i$ *iff* no solution was found, because the incoming messages were contradictory. The non-trivial fixed-points of max-product BP define quasi-solutions in 1RSB phase, and therefore define clusters $\mathbf{y}$.

Example 8 *If we initialize all messages to* 1 *for our simple 3-SAT example, the final marginals over all the variables are equal to 1, allowing all assignments for all variables. However beside this trivial fixed-point, there are other fixed points that correspond to two clusters of solutions.*

For example, considering the cluster $\{(False, False, False), (False, False, True)\}$*, the following* $\{\eta_{i\to I}\}$ *(and their corresponding* $\{\eta_{I\to i}\}$ *define a fixed-point for max-product BP:*

$$\begin{aligned}
&\eta_{1\to I}(True) = \widehat{\mu}_1(True) = 0 && \eta_{1\to I}(False) = \widehat{\mu}_1(False) = 1 \quad \forall I \in \partial 1\\
&\eta_{2\to I}(True) = \widehat{\mu}_2(True) = 0 && \eta_{2\to I}(False) = \widehat{\mu}_2(False) = 1 \quad \forall I \in \partial 2\\
&\eta_{3\to I}(True) = \widehat{\mu}_3(True) = 1 && \eta_{3\to I}(False) = \widehat{\mu}_3(False) = 1 \quad \forall I \in \partial 3
\end{aligned}$$

Here the messages indicate the allowed assignments within this particular cluster of solutions.

3.1.1 Survey Propagation

Here we define the 1RSB update equations over max-product BP messages. We skip the explicit construction of the auxiliary PGM that results in SP update equations, and confine this section to the intuition offered by SP messages. Braunstein et al. (2005) and Mézard and Montanari (2009) give details on the construction of the auxiliary-PGM. Ravanbakhsh and Greiner (2014) present an algebraic perspective on SP. Maneva et al. (2007) provide a different view on the relation of BP and SP for the satisfiability problem and Kroc et al. (2007) present empirical study of SP as applied to SAT.

Let $\mathcal{Y}_i = 2^{|\mathcal{X}_i|}$ be the power-set[8] of $\mathcal{X}_i$. Each max-product BP message can be seen as a subset of $\mathcal{X}_i$ that contains the allowed states. Therefore $\mathcal{Y}_i$ as its power-set contains all possible max-product BP messages. Each message $\boldsymbol{\nu}_{i\to I} : \mathcal{Y}_i \to [0,1]$ in the auxiliary PGM defines a *distribution* over original max-product BP messages.

Example 9 (3-COL) $\mathcal{X}_i = \{1,2,3\}$ *is the set of colors and* $\mathcal{Y}_i = \{\{\}, \{1\}, \{2\}, \{3\}, \{1,2\}, \{2,3\}, \{1,3\}, \{1,2,3\}\}$*. Here* $\mathbf{y}_i = \{\}$ *corresponds to the case where none of the colors are allowed.*

8. The power-set of $\mathcal{X}$ is the set of all subsets of $\mathcal{X}$, including $\{\}$ and $\mathcal{X}$ itself.

Applying sum-product BP to our auxiliary PGM gives entropic SP(m) updates as:

$$\boldsymbol{\nu}_{i\to I}(\mathbf{y}_i) \propto |\mathbf{y}_i|^m \sum_{\{\eta_{J\to i}\}_{J\in\partial i\setminus I}} \delta(\mathbf{y}_i, \Lambda_{i\to I}(\{\eta_{J\to i}\}_{J\in\partial i\setminus I})) \prod_{J\in\partial i\setminus I} \boldsymbol{\nu}_{J\to i}(\eta_{J\to i}) \tag{18}$$

$$\boldsymbol{\nu}_{I\to i}(\mathbf{y}_i) \propto |\mathbf{y}_i|^m \sum_{\{\eta_{j\to I}\}_{j\in\partial I\setminus i}} \delta(\mathbf{y}_i, \Lambda_{I\to i}(\{\eta_{j\to I}\}_{j\in\partial I\setminus i})) \prod_{j\in\partial I\setminus i} \boldsymbol{\nu}_{j\to I}(\eta_{j\to I}) \tag{19}$$

$$\boldsymbol{\nu}_{i\to I}(\{\}) \quad := \quad \boldsymbol{\nu}_{I\to i}(\{\}) \quad := \quad 0 \qquad \forall i,\ I \in \partial i \tag{20}$$

where the summations are over all combinations of max-product BP messages. Here the δ-function ensures that only the set of incoming messages that satisfy the original BP equations make contributions. Since we only care about the valid assignments and $\mathbf{y}_i = \{\}$ forbids all assignments, we ignore its contribution (Equation 20).

Example 10 (3-SAT) *Consider the SP message $\boldsymbol{\nu}_{1\to C_1}(\mathbf{y}_1)$ in the factor graph of Figure 1b. Here the summation in Equation* (18) *is over all possible combinations of incoming max-product BP messages $\eta_{C_2\to 1}, \ldots, \eta_{C_5\to 1}$. Since each of these messages can assume one of the three valid values—e.g., $\eta_{C^2\to 1}(x_1) \in \{\ \{True\}, \{False\}, \{True, False\}\ }$—for each particular assignment of $\mathbf{y}_1$, a total of $|\{\{True\}, \{False\}, \{True, False\}\}|^{|\partial 1\setminus C_1|} = 3^4$ possible combinations are enumerated in the summations of Equation* (18)*. However only the combinations that form a valid max-product message update have non-zero contribution in calculating $\boldsymbol{\nu}_{1\to C_1}(\mathbf{y}_1)$. These are basically the messages that appear in a max-product fixed point as discussed in Example 8.*

Each of original messages corresponds to a different sub-set of clusters and m (from Equation (13)) controls the effect of each cluster's size on its contribution. At any point, we can use these messages to estimate the marginals of $\widehat{\boldsymbol{\mu}}(\mathbf{y})$ defined in Equation (13) using

$$\widehat{\boldsymbol{\mu}}(\mathbf{y}_i) \ \propto\ |\mathbf{y}_i|^m \sum_{\{\eta_{J\to i}\}_{J\in\partial i}} \delta(\mathbf{y}_i,\ \Lambda_i(\ \{\eta_{J\to i}\}_{J\in\partial i})\ ) \prod_{J\in\partial i} \boldsymbol{\nu}_{J\to i}(\eta_{J\to i}). \tag{21}$$

This also implies a distribution over the original domain, which we slightly abuse notation to denote by

$$\widehat{\boldsymbol{\mu}}(x_i) \ \propto \sum_{\mathbf{y}_i \ni x_i} \widehat{\boldsymbol{\mu}}(\mathbf{y}_i). \tag{22}$$

The term SP usually refers to SP(0)—that is $m = 0$—where all clusters, regardless of their size, contribute the same amount to $\boldsymbol{\mu}(\mathbf{y})$. Now that we can obtain an estimate of marginals, we can employ this procedure within a decimation process to incrementally fix some variables. Here either $\widehat{\boldsymbol{\mu}}(x_i)$ or $\widehat{\boldsymbol{\mu}}(\mathbf{y}_i)$ can be used by the decimation procedure to fix the most biased variables. In the former case, a variable $\mathbf{y}_i$ is fixed to $\mathbf{y}_i^* = \{x_i^*\}$ when $x_i^* = \arg_{x_i} \max \widehat{\boldsymbol{\mu}}(x_i)$. In the latter case, $\mathbf{y}_i^* = \arg_{y_i} \max \widehat{\boldsymbol{\mu}}(\mathbf{y}_i)$. Here we use SP-dec(S) to refer to the former procedure (that uses $\widehat{\boldsymbol{\mu}}(x_i)$ to fix variables to a *single* value) and use SP-dec(C) to refer to the later case (in which variables are fixed to a *cluster* of assignments).

The original decimation procedure for κ-SAT (Braunstein et al., 2002) corresponds to SP-dec(S). SP-dec(C) for CSP with Boolean variables is only slightly different, as SP-dec(C)

can choose to fix a cluster to $\mathbf{y}_i = \{True, False\}$ in addition to the options of $\mathbf{y}_i = \{True\}$ and $\mathbf{y}_i = \{False\}$, available to SP-dec(S). However, for larger domains (*e.g.*, q-COL), SP-dec(C) has a clear advantage. For example in 3-COL, SP-dec(C) may choose to fix a cluster to $\mathbf{y}_i = \{1, 2\}$ while SP-dec(S) can only choose between $\mathbf{y}_i \in \{\{1\}, \{2\}, \{3\}\}$. This significant difference is also reflected in their comparative success-rate on q-COL.[9] (See Table 1 in Section 3.3.)

During the decimation process, usually after fixing a subset of variables, the SP marginals $\widehat{\boldsymbol{\mu}}(x_i)$ become uniform, indicating that clusters of solutions have no preference over particular assignments of the remaining variables. The same happens when we apply SP to random instances in RS phase. At this point (a.k.a. paramagnetic phase), a local search method or BP-dec can often efficiently find an assignment to the variables that are not yet fixed by decimation. Note that both SP-dec(C) and SP-dec(S) switch to local search as soon as all $\widehat{\boldsymbol{\mu}}(x_i)$ become close to uniform.

The computational complexity of each SP update of Equation (19) is $\mathcal{O}(2^{|\mathcal{X}_i|}-1)^{|\partial I|}$ as for each particular value $\mathbf{y}_i$, SP needs to consider every combination of incoming messages, each of which can take $2^{|\mathcal{X}_i|}$ values (minus the empty set). Similarly, using a naive approach the cost of update of Equation (18) is $\mathcal{O}(2^{|\mathcal{X}_i|}-1)^{|\partial i|}$. However by considering incoming messages one at a time, we can perform the same exact update in $\mathcal{O}(|\partial i|\ 2^{2|\mathcal{X}_i|})$. In comparison to the cost of BP updates, we see that SP updates are substantially more expensive for large $|\mathcal{X}_i|$ and $|\partial I|$.[10]

3.2 Perturbed Survey Propagation

The perturbation scheme that we use for SP is similar to what we did for BP. Let $\boldsymbol{\Phi}_{i\to I}(\ \{\boldsymbol{\nu}_{j\to J}\}_{j\in\Delta i,(J\in\partial i)\setminus I}\ )$ denote the update operator for the message from variable $\mathbf{y}_i$ to factor C_I. This operator is obtained by substituting Equation (19) into Equation (18) to get a single SP update equation. Let $\boldsymbol{\Phi}(\{\boldsymbol{\nu}_{i\to I}\}_{i,I\in\partial i})$ denote the aggregate SP operator, which applies $\boldsymbol{\Phi}_{i\to I}$ to update each individual message.

We perform Gibbs sampling from the "original" domain $\mathcal{X}$ using the implicit marginal of Equation (22). We denote this random operator by $\boldsymbol{\Psi} = \{\boldsymbol{\Psi}_i\}_i$, defined by

$$\boldsymbol{\nu}_{i\to I}(\mathbf{y}_i) \ = \ \boldsymbol{\Psi}_i(\ \{\boldsymbol{\nu}_{j\to J}\}_{j\in\Delta i, J\in\partial i}\ ) \ \triangleq \ \delta(\mathbf{y}_i, \{\widehat{x}_i\}) \quad \text{where } \widehat{x}_i \sim \widehat{\boldsymbol{\mu}}(x_i)$$

where the second argument of the δ-function is a singleton set, containing a sample from the estimate of marginal. Now, define the Perturbed SP operator as the convex combination of SP and either of the GS operator above:

$$\boldsymbol{\Gamma}(\{\boldsymbol{\nu}_{i\to I}\}) \ \triangleq \ \gamma\boldsymbol{\Psi}(\{\boldsymbol{\nu}_{i\to I}\}) \ + \ (1-\gamma)\boldsymbol{\Phi}(\{\boldsymbol{\nu}_{i\to I}\}).$$

Similar to perturbed BP, during iterations of Perturbed SP, γ is gradually increased from 0 to 1. If perturbed SP reaches the final iteration, the samples from the implicit

9. Previous applications of SP-dec to q-COL by Braunstein et al. (2003) used a heuristic for decimation that is similar SP-dec (C).

10. Note that our representation of distributions is over-complete—that is we are not using the fact that the distributions sum to one. However even in their more compact forms, for general CSPs, the cost of each SP update remains exponentially larger than that of BP (in $|\mathcal{X}_i|$, $|\partial I|$). However if the factors are sparse and have high order, both BP and SP allow more efficient updates.

marginals represent a satisfying assignment. The advantage of this scheme to SP-dec is that perturbed SP does not require any further local search. In fact we may apply $\mathbf{\Gamma}$ to CSP instances in the RS phase as well, where the solutions form a single giant cluster. In contrast, applying SP-dec, to these instances simply invokes the local search method.

To demonstrate this, we applied Perturbed SP(S) to benchmark CSP instances of Table 2 in which the maximum number of elements in the factor was less than 10. Here Perturbed SP(S) solved 80 instances out of 202 cases, while Perturbed BP solved 78 instances.

3.3 Experiments on random CSP

We implemented all the methods above for general factored CSP using the libdai code base (Mooij, 2010). To our knowledge this is the first general implementation of SP and SP-dec. Previous applications of SP-dec to κ-SAT and q-COL (Braunstein et al., 2003; Mulet et al., 2002; Braunstein et al., 2002) were specifically tailored to just one of those problems.

Here we report the results on κ-SAT for $\kappa \in \{3, 4\}$ and q-COL for $q \in \{3, 4, 9\}$. We used the procedure discussed in the examples of Section 3 to produce 100 random instances with $N = 5,000$ variables for each control parameter α. We report the probability of finding a satisfying assignment for different methods (*i.e.*, the portion of 100 instances that were satisfied by each method). For coloring instances, to help decimation, we break the initial symmetry of the problem by fixing a single variable to an arbitrary value.

For BP-dec and SP-dec, we use a convergence threshold of $\epsilon = .001$ and fix $\rho = 1\%$ of variables per iteration of decimation. Perturbed BP and Perturbed SP use $T = 1000$ iterations. Decimation-based methods use a maximum of $T = 1000$ iterations per iteration of decimation. If any of the methods failed to find a solution in the first attempt, T was increased by a factor of 4 at most 3 times (so in the final attempt: $T = 64,000$). To avoid blow-up in run-time, for BP-dec and SP-dec, only the maximum iteration, T, during the first iteration of decimation, was increased (this is similar to the setting of Braunstein et al. (2002) for SP-dec). For both variations of SP-dec (see Section 3.1.1), after each decimation step, if $\max_{i,x_i} \mu(x_i) - \frac{1}{|\mathcal{X}_i|} < .01$ we consider the instance para-magnetic, and run BP-dec (with $T = 1000$, $\epsilon = .001$ and $\rho = 1\%$) on the simplified instance.

Figure 4(first row) visualizes the success rate of different methods on 100 instances of 3-SAT (right) and 3-COL (left). Figure 4(second row) reports the number of variables that are fixed by SP-dec(C) and (S) before calling BP-dec as local search. The third row shows the average amount of time that is used to find a satisfying solution. This does not include the failed attempts. For SP-dec variations, this time includes the time used by local search. The final row of Figure 4 shows the number of iterations used by each method at each level of difficulty over the successful instances. Here the area of each disk is proportional to the frequency of satisfied instances with that particular number of iterations for each control parameter and inference method[11].

Here we make the following observations:

- **Perturbed BP is much more effective than BP-dec, while remaining ten to hundreds of times more efficient.**

11. The number of iterations are rounded to the closest power of two.

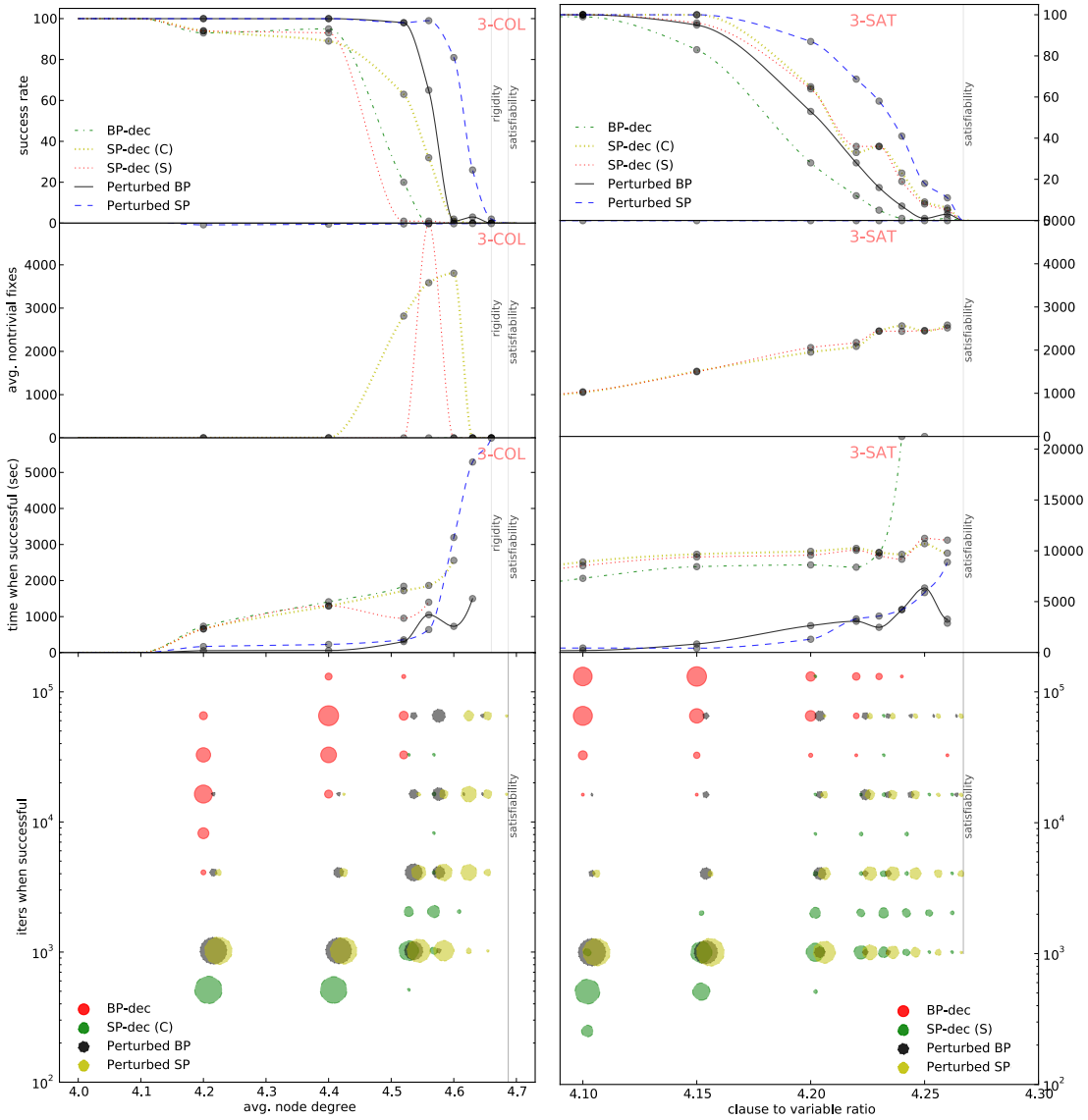


Figure 4: (first row) Success-rate of different methods for 3-COL and 3-SAT for various control parameters. (second row) The average number of variables (out of $N = 5000$) that are fixed using SP-dec (C) and (S) before calling local search, averaged over 100 instances. (third row) The average amount of time (in seconds) used by the successful setting of each method to find a satisfying solution. For SP-dec(C) and (S) this includes the time used by local search. (fourth row) The number of iterations used by different methods at different control parameters, when the method was successful at finding a solution. The number of iterations for each of 100 random instances is rounded to the closest power of 2. This does not include the iterations used by local search after SP-dec.

- As the control parameter grows larger, the chance of requiring more iterations to satisfy the instance increases for all methods.
- Although computationally very inefficient, BP-dec is able to find solutions for instances with larger control parameters than suggested by previous results (*e.g.*, Mézard and Montanari, 2009).
- For many instances where SP-dec(C) and (S) use few iterations, the variables are fixed to a trivial cluster $\mathbf{y}_i = \mathcal{X}_i$, which allows all assignments. This is particularly pronounced for 3-COL, where up to $\alpha = 4.4$ the non-trivial fixes remains zero and therefore the success rate up to this point is solely due to BP-dec.
- While for 3-SAT, SP-dec(C) and SP-dec(S) have a similar performance, for 3-COL, SP-dec(C) significantly outperforms SP-dec(S).

Table 1 reports the success-rate as well as the average of total iterations in the *successful* attempts of each method. Here the number of iterations for SP-dec(C) and (S) is the sum of iterations used by the method and the following local search. We observe that Perturbed BP can solve most of the easier instances using only $T = 1000$ iterations (*e.g.*, see Perturb BP's result for 3-SAT at $\alpha = 4.1$, 3-COL at $\alpha = 4.2$ and 9-COL at $\alpha = 33.4$).

Table 1 also supports our speculation in Section 3.1.1 that **SP-dec(C) is in general preferable to SP-dec(S)**, in particular when applied to the coloring problem.

The most important advantage of Perturbed BP over SP-dec and Perturbed SP is that Perturbed BP can be applied to instances with large factor cardinality (*e.g.*, 10-SAT) and large variable domains (*e.g.*, 9-COL). For example for 9-COL, the cardinality of each SP message is $2^9 = 512$, which makes SP-dec and Perturbed SP impractical. Here BP-dec is not even able to solve a single instance around the dynamical transition (as low as $\alpha = 33.4$) while Perturbed BP satisfies all instances up to $\alpha = 34.1$.[12] Besides the experimental results reported here, we have also used perturbed BP to efficiently solve other CSPs such as K-Packing, K-set-cover and clique-cover within the context of min-max inference (Ravanbakhsh et al., 2014).

3.4 Discussion

It is easy to check that, for $m = 1$, SP updates produce sum-product BP messages as an average case; that is, the SP updates (equations 18, 19) reduce to that of sum-product BP (equations 3, 4) where

$$\nu_{i\to I}(x_i) \quad \propto \sum_{\mathbf{y}_i \ni x_i} \boldsymbol{\nu}_{i\to I}(\mathbf{y}_i)$$

This suggests that the BP equation remains correct wherever SP(1) holds, which has lead to the belief that BP-dec should perform well up to the condensation transition (Krzakala et al., 2007). However in reaching this conclusion, the effect of decimation was ignored. More

12. Note that for 9-COL condensation transition happens after rigidity transition. So if we were able to find solutions after rigidity, it would have implied that condensation transition marks the onset of difficulty. However, this did not occur and similar to all other cases, Perturbed BP failed before rigidity transition.

Problem	ctrl param α	BP-dec		SP-dec(C)		SP-dec(S)		Perturbed BP		Perturbed SP	
		avg. iters.	success rate	avg. iters.	success rate	avg. iters.	success rate	avg. iters.	success rate	avg. iters.	success rate
3-SAT	3.86	dynamical and condensation transition									
	4.1	85405	99%	102800	100%	96475	100%	1301	100%	1211	100%
	4.15	104147	83%	118852	100%	111754	96%	5643	95%	1121	100%
	4.2	93904	28%	118288	65%	113910	64%	19227	53%	3415	87%
	4.22	100609	12%	112910	33%	114303	36%	22430	28%	8413	69%
	4.23	123318	5%	109659	36%	107783	36%	18438	16%	9173	58%
	4.24	165710	1%	126794	23%	118284	19%	29715	7%	10147	41%
	4.25	N/A	0%	123703	9%	110584	8%	64001	1%	14501	18%
	4.26	37396	1%	83231	6%	106363	5%	32001	3%	22274	11%
	4.268	satisfiability transition									
4-SAT	9.38	dynamical transition									
	9.547	condensation transition									
	9.73	134368	8%	119483	32%	120353	35%	25001	43%	11142	86%
	9.75	168633	5%	115506	15%	96391	21%	36668	27%	9783	68%
	9.78	N/A	0%	83720	9%	139412	7%	34001	12%	11876	37%
	9.88	rigidity transition									
	9.931	satisfiability transition									
3-COL	4	dynamical and condensation transition									
	4.2	24148	93%	25066	94%	24634	94%	1511	100%	1151	100%
	4.4	51590	95%	52684	89%	54587	93%	1691	100%	1421	100%
	4.52	61109	20%	68189	63%	54736	1%	7705	98%	2134	98%
	4.56	N/A	0%	63980	32%	13317	1%	28047	65%	3607	99%
	4.6	N/A	0%	74550	2%	N/A	0%	16001	1%	18075	81%
	4.63	N/A	0%	N/A	0%	N/A	0%	48001	3%	29270	26%
	4.66	rigidity transition									
	4.66	N/A	0%	N/A	0%	N/A	0%	N/A	0%	40001	2%
	4.687	satisfiability transition									
4-COL	8.353	dynamical transition									
	8.4	64207	92%	72359	88%	71214	93%	1931	100%	1331	100%
	8.46	dynamical transition									
	8.55	77618	13%	60802	13%	62876	9%	3041	100%	5577	100%
	8.7	N/A	0%	N/A	0%	N/A	0%	50287	14%	N/A	0%
	8.83	rigidity transition									
	8.901	satisfiability transition									
9-COL	33.45	dynamical transition									
	33.4	N/A	0%	N/A	N/A	N/A	N/A	1061	100%	N/A	N/A
	33.9	N/A	0%	N/A	N/A	N/A	N/A	3701	100%	N/A	N/A
	34.1	N/A	0%	N/A	N/A	N/A	N/A	12243	100%	N/A	N/A
	34.5	N/A	0%	N/A	N/A	N/A	N/A	48001	6%	N/A	N/A
	35.0	N/A	0%	N/A	N/A	N/A	N/A	N/A	0%	N/A	N/A
	39.87	rigidity transition									
	43.08	condensation transition									
	43.37	satisfiability transition									

Table 1: Comparison of different methods on $\{3, 4\}$-SAT and $\{3, 4, 9\}$-COL. For each method the success-rate and the average number of iterations (including local search) on successful attempts are reported. The approximate location of phase transitions are given by Montanari et al. (2008); Zdeborova and Krzakala (2007).

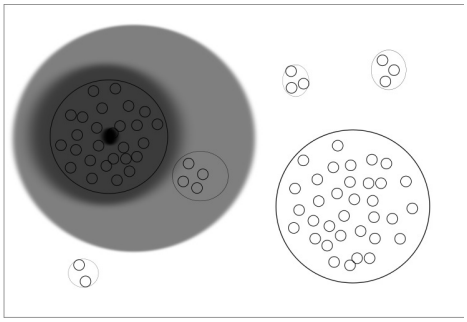


Figure 5: This schematic view demonstrates the clustering during condensation phase. Here assume horizontal and vertical axes correspond to x_1 and x_2. Considering the whole space of assignments, x_1 and x_2 are highly correlated. The formation of this correlation between distant variables on a PGM breaks BP. Now assume that Perturbed BP messages are focused on the largest shaded ellipse. In this case the correlation is significantly reduced.

recent analyses (Coja-Oghlan, 2011; Montanari et al., 2007; Ricci-Tersenghi and Semerjian, 2009) draw a similar conclusion about the effect of decimation: At some point during the decimation process, variables form long-range correlations such that fixing one variable may imply an assignment for a portion of variables that form a loop, potentially leading to contradictions. Alternatively the same long-range correlations result in BP's lack of convergence and error in marginals that may lead to unsatisfying assignments.

Perturbed BP avoids the pitfalls of BP-dec in two ways: **(I)** Since many configurations have non-zero probability until the final iteration, Perturbed BP can avoid contradictions by adapting to the most recent choices. This is in contrast to decimation in which variables are fixed once and are unable to change afterwards. A backtracking scheme suggested by Parisi (2003) attempts to fix the same problem with SP-dec. **(II)** We speculate that simultaneous bias of all messages towards sub-regions over which the BP equations remain valid, prevents the formation of long-range correlations between variables that breaks BP in 1sRSB; see Figure 5.

In all experiments, we observed that Perturbed BP is competitive with SP-dec, while BP-dec often fails on much easier problems. We saw that the cost of each SP update grows exponentially faster than the cost of each BP update. Meanwhile, our perturbation scheme adds a negligible cost to that of BP—*i.e.*, that of sampling from these local marginals and updating the outgoing messages accordingly. Considering the computational complexity of SP-dec, and also the limited setting under which it is applicable, Perturbed BP is an attractive substitute. Furthermore our experimental results also suggest that Perturbed SP(S) is a viable option for real-world CSPs with small variable domains and constraint cardinalities.

4. Conclusion

We considered the challenge of efficiently producing assignments that satisfy hard combinatorial problems, such as κ-SAT and q-COL. We focused on ways to use message passing methods to solve CSPs, and introduced a novel approach, Perturbed BP, that combines BP and GS in order to sample from the set of solutions. We demonstrated that Perturbed BP is significantly more efficient and successful than BP-dec. We also demonstrated that Perturbed BP can be as powerful as a state-of-the-art algorithm (SP-dec), in solving rCSPs while remaining tractable for problems with large variable domains and factor cardinalities. Furthermore we provided a method to apply the similar perturbation procedure to SP, producing the Perturbed SP process that outperforms SP-dec in solving difficult rCSPs.

Acknowledgments

We would like to thank anonymous reviewers for their constructive and insightful feedback. RG received support from NSERC and Alberta Innovate Center for Machine Learning (AICML). Research of SR has been supported by Alberta Innovates Technology Futures, AICML and QEII graduate scholarship. This research has been enabled by the use of computing resources provided by WestGrid and Compute/Calcul Canada.

References

D. Achioptas and G. Sorkin. Optimal myopic algorithms for random 3-SAT. In *Proceedings of 41st Annual Symposium on Foundations of Computer Science*, pages 590–600. IEEE, 2000.

D. Achlioptas and A. Coja-Oghlan. Algorithmic barriers from phase transitions. *arXiv:0803.2122*, March 2008.

D. Achlioptas, A. Naor, and Y. Peres. Rigorous location of phase transitions in hard optimization problems. *Nature*, 435(7043):759–764, June 2005. ISSN 0028-0836.

C. Andrieu, N. de Freitas, A. Doucet, and M. Jordan. An introduction to MCMC for machine learning. *Machine Learning*, 2003.

A. Braunstein, M. Mézard, and R. Zecchina. Survey propagation: an algorithm for satisfiability. *Random Structures and Algorithms*, 27(2):19, 2002.

A. Braunstein, R. Mulet, A. Pagnani, M. Weigt, and R. Zecchina. Polynomial iterative algorithms for coloring and analyzing random graphs. *Physical Review E*, 68(3):036702, 2003.

A. Braunstein, M. Mézard, M. Weigt, and R. Zecchina. Constraint satisfaction by survey propagation. *Computational Complexity and Statistical Physics*, page 107, 2005.

P. Cheeseman, B. Kanefsky, and W. Taylor. Where the really hard problems are. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence - Volume*

1, IJCAI'91, pages 331–337, San Francisco, CA, USA, 1991. Morgan Kaufmann Publishers Inc. ISBN 1-55860-160-0.

S. Cocco, O. Dubois, J. Mandler, and R. Monasson. Rigorous decimation-based construction of ground pure states for spin-glass models on random rattices. *Physical review letters*, 90(4):047205, 2003.

A. Coja-Oghlan. On belief propagation guided decimation for random k-SAT. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '11, pages 957–966. SIAM, 2011.

V. Deolalikar. Deolalikar P vs NP paper, 2010. URL `http://michaelnielsen.org/polymath1/index.php?title=Deolalikar_P_vs_NP_paper`.

Y. Fu and P. Anderson. Application of statistical mechanics to NP-Complete problems in combinatorial optimisation. *Journal of Physics A: Mathematical and General*, 19(9): 1605–1620, June 1986. ISSN 0305-4470, 1361-6447.

M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*, volume 44 of *A Series of Books in the Mathematical Sciences*. Freeman W., 1979. ISBN 0716710455.

A. Ihler and D. Mcallester. Particle belief propagation. In *International Conference on Artificial Intelligence and Statistics*, pages 256–263, 2009.

S. Janson, T. Luczak, and V. Kolchin. Random graphs. *Bulletin of the London Mathematical Society*, 33:363–383, 2001.

K. Kask, R. Dechter, and V. Gogate. Counting-based look-ahead schemes for constraint satisfaction. In *Principles and Practice of Constraint Programming*, pages 317–331. Springer, 2004.

H. Kautz, A. Sabharwal, and B. Selman. Incomplete algorithms. In *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*, pages 185–203. IOS Press, 2009. ISBN 978-1-58603-929-5.

S. Kirkpatrick and B. Selman. Critical behavior in the satisfiability of random boolean expressions. *Science*, 264(5163):1297–1301, 1994.

L. Kroc, A. Sabharwal, and B. Selman. Survey propagation revisited. In *Proceedings of the 23rd conference on Uncertainty in Artificial Intelligence*, pages 217–226, 2007.

L. Kroc, A. Sabharwal, and B. Selman. Message-passing and local heuristics as decimation strategies for satisfiability. In *Proceedings of the 2009 ACM Symposium on Applied Computing*, pages 1408–1414. ACM, 2009.

F. Krzakala, A. Montanari, F. Ricci-Tersenghi, G. Semerjian, and L. Zdeborová. Gibbs states and the set of solutions of random constraint satisfaction problems. *Proceedings of the National Academy of Sciences*, 104(25):10318–10323, 2007.

F. Kschischang, B. Frey, and H. Loeliger. Factor graphs and the sum-product algorithm. *Information Theory, IEEE Transactions on*, 47(2):498–519, 2001.

Ch. Lecoutre. A collection of CSP benchmark instances., October 2013. URL `http://www.cril.univ-artois.fr/~lecoutre/benchmarks.html`.

E. Maneva, E. Mossel, and M. Wainwright. A new look at survey propagation and its generalizations. *Journal of the ACM*, 54(4):17, 2007.

M. Mézard and A. Montanari. *Information, Physics, and Computation*. Oxford, 2009.

M. Mézard and G. Parisi. The Bethe lattice spin glass revisited. *The European Physical Journal B-Condensed Matter and Complex Systems*, 20(2):217–233, 2001.

M. Mézard and G. Parisi. The cavity method at zero temperature. *Journal of Statistical Physics*, page 111, 2003.

M. Mézard, G. Parisi, and M. Virasoro. *Spin Glass Theory and Beyond*. Singapore: World Scientific, 1987.

M. Mézard, G. Parisi, and R. Zecchina. Analytic and algorithmic solution of random satisfiability problems. *Science*, 297(5582):812–815, August 2002. ISSN 0036-8075, 1095-9203.

D. Mitchell, B. Selman, and H. Levesque. Hard and easy distributions of SAT problems. In *Association for the Advancement of Artificial Intelligence (AAAI)*, volume 92, pages 459–465, 1992.

A. Montanari, F. Ricci-tersenghi, and G. Semerjian. Solving constraint satisfaction problems through belief propagation-guided decimation. In *in Proceedings of the Allerton Conference on Communication, Control, and Computing*, 2007.

A. Montanari, F. Ricci-Tersenghi, and G. Semerjian. Clusters of solutions and replica symmetry breaking in random k-satisfiability. *Journal of Statistical Mechanics*, page 04004, 2008.

J. Mooij. libDAI: A free and open source C++ library for discrete approximate inference in graphical models. *Journal of Machine Learning Research*, 11:2169–2173, August 2010.

R. Mulet, A. Pagnani, M. Weigt, and R. Zecchina. Coloring random graphs. *Physical Review Letters*, 89(26):268701, 2002.

N. Noorshams and M. Wainwright. Belief propagation for continuous state spaces: Stochastic message-passing with quantitative guarantees. *Journal of Machine Learning Research*, 14:2799–2835, 2013.

G. Parisi. A backtracking survey propagation algorithm for k-satisfiability. *Arxiv preprint Condmat:0308510*, page 9, 2003.

J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, volume 88 of *Representation and Reasoning*. Morgan Kaufmann, 1988. ISBN 1558604790.

S. Ravanbakhsh and R. Greiner. Revisiting algebra and complexity of inference in graphical models. *arXiv:cs/1409.7410v4*, 2014.

S. Ravanbakhsh, C. Srinivasa, B. Frey, and R. Greiner. Min-max problems on factor-graphs. *Proceedings of The 31st International Conference on Machine Learning*, pages 1035–1043, 2014.

F. Ricci-Tersenghi. Being glassy without being hard to solve. *Science*, 330(6011):1639–1640, 2010.

F. Ricci-Tersenghi and G. Semerjian. On the cavity method for decimated random constraint satisfaction problems and the analysis of belief propagation guided decimation algorithms. *Journal of Statistical Mechanics*, page 9001, April 2009.

Ch. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer Texts in Statistics. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005. ISBN 0387212396.

O. Roussel and Ch. Lecoutre. XML representation of constraint networks: Format XCSP 2.1. *arXiv:0902.2362*, 2009.

B. Selman, H. Kautz, and B. Cohen. Noise strategies for improving local search. In *Association for the Advancement of Artificial Intelligence*, volume 94, pages 337–343, 1994.

L. G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189–201, 1979.

L. Zdeborova and F. Krzakala. Phase transitions in the coloring of random graphs. *Physical Review E*, 76(3):031131, 2007.

Appendix A. Detailed Results for Benchmark CSP

In Table 2 we report the average number of iterations and average time for the attempt that successfully found a satisfying assignment (*i.e.*, the failed instances are not included in the average). We also report the number of satisfied instances for each method as well as the number of satisfiable instance in that series of problems (if known). Further information about each data-set maybe obtained from Lecoutre (2013).

				BP-dec			Perturbed BP		
problem	series	instances	# satisfiable	# satisfied	avg. time (s)	avg. iters	# satisfied	avg. time (s)	avg. iters
Geometric	-	100	92	77	208.63	30383	81	.70	74
Dimacs	aim-50	24	16	9	11.41	25344	14	.07	181
	aim-100	24	16	8	18.2	16755	11	.15	213
	aim-200	24	N/A	7	401.90	160884	6	.17	46
	ssa	8	N/A	4	.60	373.25	4	.50	86
	jhnSat	16	16	16	5839.86	141852	13	9.82	117
	varDimacs	9	N/A	4	2.95	715	4	.12	18
QCP	QCP-10	15	10	10	43.87	30054	10	.22	51
	QCP-15	15	10	3	5659.70	600741	4	9.59	530
	QCP-25	15	10	0	0	0	0	0	0
Graph-Coloring	ColoringExt	17	N/A	4	.05	103	5	.04	25
	school	8	N/A	0	N/A	N/A	5	62.86	153
	myciel	16	N/A	5	.21	59	5	.05	11
	hos	13	N/A	5	27.34	606	5	10.04	37
	mug	8	N/A	4	.068	313	4	.004	11
	register-fpsol	25	N/A	0	N/A	N/A	0	N/A	N/A
	register-inithx	25	N/A	0	N/A	N/A	0	N/A	N/A
	register-zeroin	14	N/A	3	5906.16	26544	0	N/A	N/A
	register-mulsol	49	N/A	5	59.27	418	0	N/A	N/A
	sgb-queen	50	N/A	7	35.66	916	11	7.56	81
	sgb-games	4	N/A	1	.91	434	1	.07	21
	sgb-miles	34	N/A	4	20.86	371	2	4.20	181
	sgb-book	26	N/A	5	1.72	444	5	.18	39
	leighton-5	8	N/A	0	N/A	N/A	0	N/A	N/A
	leighton-15	28	N/A	0	N/A	N/A	1	106.46	641
	leighton-25	29	N/A	2	304.49	1516	2	94.11	241
All Interval Series	series	12	12	2	4.78	11319	7	1.85	520
Job Shop	e0ddr1	10	10	9	707.74	9195	5	37	257
	e0ddr2	10	10	5	3640.40	26544	7	74.49	366
	ewddr2	10	10	10	10871.96	48053	9	21.24	72
Schurr's Lemma	-	10	N/A	1	39.89	120152	2	.97	100
Ramsey	Ramsey 3	8	N/A	1	.01	61	4	.75	283
	Ramsey 4	8	N/A	2	12941.51	561300	7	7.39	81
Chessboard Coloration	-	14	N/A	5	35.51	3111	5	.66	27
Hanoi	-	3	3	3	.48	12	3	.52	14
Golomb Ruler	Arity 3	8	N/A	2	1.39	103	2	19.78	660
Queens	queens	8	8	7	3.30	159	8	2.43	57
Multi-Knapsack	mknap	2	2	2	2.44	6	2	4.41	10
Driver	-	7	7	5	10.14	1438	5	4.74	274
Composed	25-10-20	10	10	8	1.62	695	5	.17	38
Langford	lagford-ext	4	2	0	N/A	N/A	1	.002	10
	lagford 2	22	N/A	4	.67	127	10	11.64	10
	lagford 3	20	N/A	0	N/A	N/A	N/A	N/A	N/A

Table 2: Comparison of Perturbed BP and BP-guided decimation on benchmark CSPs.

Journal of Machine Learning Research 16 (2015) 1275-1304 Submitted 12/12; Revised 3/15; Published 7/15

Learning Sparse Low-Threshold Linear Classifiers

Sivan Sabato SABATOS@CS.BGU.AC.IL
Ben-Gurion University of the Negev
Beer Sheva, 8410501, Israel

Shai Shalev-Shwartz SHAIS@CS.HUJI.AC.IL
Benin School of Computer Science and Engineering
The Hebrew University
Givat Ram, Jerusalem 91904, Israel

Nathan Srebro NATI@TTIC.EDU
Toyota Technological Institute at Chicago
6045 S. Kenwood Ave.
Chicago, IL 60637

Daniel Hsu DJHSU@CS.COLUMBIA.EDU
Department of Computer Science
Columbia University
1214 Amsterdam Avenue, #0401
New York, NY 10027

Tong Zhang TZHANG@STAT.RUTGERS.EDU
Department of Statistics
Rutgers University
Piscataway, NJ 08854

Editor: Koby Crammer

Abstract

We consider the problem of learning a non-negative linear classifier with a ℓ_1-norm of at most k, and a fixed threshold, under the hinge-loss. This problem generalizes the problem of learning a k-monotone disjunction. We prove that we can learn efficiently in this setting, at a rate which is linear in both k and the size of the threshold, and that this is the best possible rate. We provide an efficient online learning algorithm that achieves the optimal rate, and show that in the batch case, empirical risk minimization achieves this rate as well. The rates we show are tighter than the uniform convergence rate, which grows with k^2.

Keywords: linear classifiers, monotone disjunctions, online learning, empirical risk minimization, uniform convergence

1. Introduction

We consider the problem of learning non-negative, low-ℓ_1-norm linear classifiers *with a fixed (or bounded) threshold*. That is, we consider hypothesis classes over instances $x \in [0,1]^d$ of the following form:

$$\mathcal{H}_{k,\theta} = \left\{ x \mapsto \langle w, x \rangle - \theta \;\middle|\; w \in \mathbb{R}^d_+, \|w\|_1 \leq k \right\}, \tag{1}$$

where we associate each (real valued) linear predictor in $\mathcal{H}_{k,\theta}$ with a binary classifier:[1]

$$x \mapsto \text{sign}(\langle w, x\rangle - \theta) = \begin{cases} 1 & \text{if } \langle w, x\rangle > \theta \\ -1 & \text{if } \langle w, x\rangle < \theta \end{cases}. \tag{2}$$

Note that the hypothesis class is specified by both the ℓ_1-norm constraint k *and* the fixed threshold θ. In fact, the main challenge here is to understand how the complexity of learning $\mathcal{H}_{k,\theta}$ changes with θ.

The classes $\mathcal{H}_{k,\theta}$ can be seen as a generalization and extension of the class of k-monotone-disjunctions and r-of-k-formulas. Considering binary instances $x \in \{0,1\}^d$, the class of k-monotone-disjunctions corresponds to linear classifiers with binary weights, $w \in \{0,1\}^d$, with $\|w\|_1 \leq k$ and a fixed threshold of $\theta = \frac{1}{2}$. That is, a restriction of $\mathcal{H}_{k,\frac{1}{2}}$ to integer weights and integer instances. More generally, the class of r-of-k formulas (i.e., formulas which are true if at least r of a specified k variables are true) corresponds to a similar restriction, but with a threshold of $\theta = r - \frac{1}{2}$.

Studying k-disjunctions and r-of-k formulas, Littlestone (1988) presented the efficient Winnow online learning rule, which admits an online mistake bound (in the separable case) of $O(k \log d)$ for k-disjunctions and $O(rk \log d)$ for r-of-k-formulas. In fact, in this analysis, Littlestone considered also the more general case of real-valued weights, corresponding to the class $\mathcal{H}_{k,\theta}$ over binary instances $x \in \{0,1\}^d$ and for separable data, and showed that Winnow enjoys a mistake bound of $O(\theta k \log d)$ in this case as well. By applying a standard online-to-batch conversion (see, e.g., Shalev-Shwartz, 2012), one can also achieve a sample complexity upper bound of $O(\theta k \log(d)/\epsilon)$ for batch supervised learning of this class in the separable case.

In this paper, we consider the more general case, where the instances x can also be fractional, i.e., where $x \in [0,1]^d$ and in the agnostic, non-separable, case. It should be noted that Littlestone (1989) also studied a limited version of the non-separable setting.

In order to move on to the fractional and agnostic analysis, we must clarify the loss function we will use, and the related issue of separation with a margin. When the instances x and weight vectors w are integer-valued, we have that $\langle w, x\rangle$ is always integer. Therefore, if positive and negative instances are at all separated by some predictor w (i.e., $\text{sign}(\langle w, x\rangle - \theta) = y$ where $y \in \{\pm 1\}$ denotes the target label), they are necessarily separated by a margin of half. That is, setting $\theta = r - \frac{1}{2}$ for an integer r, we have $y(\langle w, x\rangle - \theta) \geq \frac{1}{2}$. Moving to fractional instances and weight vectors, we need to require such a margin explicitly. And if considering the agnostic case, we must account not only for misclassified points, but also for margin violations. As is standard both in online learning (e.g., the agnostic Perceptron guarantee of Gentile 2003) and in statistical learning using convex optimization (e.g., support vector machines), we will rely on the hinge loss at margin half,[2] which is equal to: $2 \cdot \left[\frac{1}{2} - yh(x)\right]_+$. The hinge loss is a convex upper bound to the zero-one loss (that is, the misclassification rate) and so obtaining learning guarantees for it translates to guarantees on the misclassification error rate.

1. The value of the mapping when $\langle w, x\rangle = \theta$ can be arbitrary, as our results and our analysis do not depend on it.
2. Measuring the hinge loss at a margin of half rather than a margin of one is an arbitrary choice, which corresponds to a scaling by a factor of two, which fits better with the integer case discussed above.

Phrasing the problem as hinge-loss minimization over the hypothesis class $\mathcal{H}_{k,\theta}$, we can use Online Exponentiated Gradient (EG) (Kivinen and Warmuth, 1994) or Online Mirror Descent (MD) (e.g., Shalev-Shwartz, 2007; Srebro et al., 2011), which rely only on the ℓ_1-bound and hold for any threshold. In the statistical setting, we can use Empirical Risk Minimization (ERM), in this case minimizing the empirical hinge loss, and rely on uniform concentration for bounded ℓ_1 predictors (Schapire et al., 1997; Zhang, 2002; Kakade et al., 2009), again regardless of the threshold.

However, these approaches yield mistake bounds or sample complexities that scale quadratically with the ℓ_1 norm, that is with k^2 rather than with θk. Since the relevant range of thresholds is $0 \leq \theta \leq k$, a scaling of θk is always better than k^2. When θ is large, that is, roughly $k/2$, the Winnow bound agrees with the EG and MD bounds. But when we consider classification with a small threshold (for instance, $\theta = \frac{1}{2}$) in the case of disjunctions, the Winnow analysis clarifies that this is a much simpler class, with a resulting smaller mistake bound and sample complexity, scaling with k rather than with k^2. This distinction is lost in the EG and MD analyses, and in the ERM guarantee based on uniform convergence arguments. For small thresholds, where $\theta = O(1)$, the difference between these analyses and the Winnow guarantee is a factor of k.

Our starting point and our main motivation for this paper is to understand this gap between the EG, MD and uniform concentration analyses and the Winnow analysis. Is this gap an artifact of the integer domain or the separability assumption? Or can we obtain guarantees that scale as θk rather then k^2 also in the non-integer non-separable case? In the statistical setting, must we use an online algorithm (such as Winnow) and an online-to-batch conversion in order to ensure a sample complexity that scales with θk, or can we obtain the same sample complexity also with ERM? This is an important question, since the ERM algorithm is considered the canonical batch learning algorithm, and understanding its scope and limitations is of theoretical and practical interest. A related question is whether it is possible to establish uniform convergence guarantees with a dependence on θk rather then k^2, or do the learning guarantees here arise from a more delicate argument.

If an ERM algorithm obtains similar bounds to the ones of the online algorithm with online-to-batch convergence, then any algorithm that can minimize the risk on the sample can be used for learning in this setting. Moreover, this advances our theoretical understanding of the limitations and scope of the canonical ERM algorithm.

The gap between the Winnow analysis and the more general ℓ_1-norm-based analyses is particularly interesting since we know that, in a sense, online mirror descent always provides the best possible rates in the online setting (Srebro et al., 2011). It is thus desirable to understand whether mirror descent is required here to achieve the best rates, or can it be replaced by a simple regularized loss minimization.

Answering the above questions, our main contributions are:

- We provide a variant of online Exponentiated Gradient, for which we establish a regret bound of $O(\sqrt{\theta k \log(d) T})$ for $\mathcal{H}_{k,\theta}$, improving on the $O(\sqrt{k^2 \log(d) T})$ regret guarantee ensured by the standard EG analysis. We do so using a more refined analysis based on local norms. Using a standard online-to-batch conversion, this yields a sample complexity of $O(\theta k \log(d)/\epsilon^2)$ in the statistical setting. This result is given in Corollary 5, Section 3.

- In the statistical agnostic PAC setting, we show that the rate of uniform convergence of the empirical hinge loss of predictors in $\mathcal{H}_{k,\theta}$ is indeed $\Omega(\sqrt{k^2/m})$ where m is the sample size, corresponding to a sample complexity of $\Omega(k^2/\epsilon^2)$, even when θ is small. We show this in Theorem 21 in Section 5. Nevertheless, we establish a learning guarantee for empirical risk minimization which matches the online-to-batch guarantee above (up to logarithmic factors), and ensures a sample complexity of $\tilde{O}(\theta k \log(d)/\epsilon^2)$ also when using ERM. This is obtained by a more delicate local analysis, focusing on predictors which might be chosen as empirical risk minimizers, rather than a uniform analysis over the entire class $\mathcal{H}_{k,\theta}$. The result is given in Theorem 6, Section 4.

- We also establish a matching lower bound (up to logarithmic factors) of $\Omega(\theta k/\epsilon^2)$ on the required sample complexity for learning $\mathcal{H}_{k,\theta}$ in the statistical setting. This shows that our ERM analysis is tight (up to logarithmic factors), and that, furthermore, the regret guarantee we obtain in the online setting is likewise tight up to logarithmic factors. This lower bound is provided in Theorem 17, Section 5.

1.1 Related Prior Work

We discussed Littlestone's work on Winnow at length above. In our notation, Littlestone (1988) established a mistake bound (that is, a regret guarantee in the separable case, where there exists a predictor with zero hinge loss) of $O(k\theta \log(d))$ for $\mathcal{H}_{k,\theta}$, when the instances are integer $x \in \{0,1\}^d$. Littlestone also established a lower bound of $k \log(d/k)$ on the VC-dimension of k-monotone-disjunctions, corresponding to the case $\theta = \frac{1}{2}$, thus implying a $\Omega(k \log(d/k)/\epsilon^2)$ lower bound on learning $\mathcal{H}_{k,\frac{1}{2}}$. However, the question of obtaining a lower bound for other values of the threshold θ was left open by Littlestone.

In the agnostic case, Auer and Warmuth (1998) studied the discrete problem of k-monotone disjunctions, corresponding to $\mathcal{H}_{k,\frac{1}{2}}$ with integer instances $x \in \{0,1\}^d$ and integer weights $w \in \{0,1\}^d$, under the *attribute loss*, defined as the number of variables in the assignment that need to be flipped in order to make the predicted label correct. They provide an online algorithm with an expected mistake bound of $A^* + 2\sqrt{A^* k \ln(d/k)} + O(k \ln(d/k))$, where A^* is the best possible attribute loss for the given online sequence. An online-to-batch conversion thus achieves here a zero-one loss which converges to the optimal attribute loss on this problem at the rate of $O(k \ln(d/k)/\epsilon^2)$. Since the attribute loss is upper bounded by the hinge loss, a similar result, in which A^* is replaced with the optimal hinge-loss for the given sequence, also holds for the same algorithm. This establishes an agnostic guarantee of the desired form, for a threshold of $\theta = \frac{1}{2}$, and when both the instances and weight vectors are integers.

2. Notations and Definitions

For a real number q, we denote its positive part by $[q]_+ := \max\{0, q\}$. We denote universal positive constants by C. The value of C may be different between statements or even between lines of the same expression. We denote by $\mathbb{R}^d_+$ the non-negative orthant in $\mathbb{R}^d$. The all-zero vector in $\mathbb{R}^d$ is denoted by $\mathbf{0}$. For an integer n, we denote $[n] = \{1, \ldots, n\}$. For a vector $x \in \mathbb{R}^d$, and $i \in [d]$, $x[i]$ denotes the i'th coordinate of x.

We will slightly overload notation and use $\mathcal{H}_{k,\theta}$ to denote both the set of linear predictors $x \mapsto \langle w, x\rangle - \theta$ and the set of vectors $w \in \mathbb{R}^d_+$ such that $\|w\|_1 \leq k$. We will use w to denote both the vector and the linear predictor associated with it.

For convenience we will work with *half* the hinge loss at margin half, and denote this loss, for a predictor $w \in \mathcal{H}_{k,\theta}$, for $\theta \in [0, k]$, by

$$\ell_\theta(x, y, w) := \left[\frac{1}{2} - y(\langle w, x\rangle - \theta)\right]_+.$$

The subscript θ will sometimes be omitted when it is clear from context. We term ℓ_θ the *Winnow loss*.

Echoing the half-integer thresholds for k-monotone-disjunctions, r-of-k formulas, and the discrete case more generally, we will denote $r = \theta + \frac{1}{2}$, so that $\theta = r - \frac{1}{2}$. In the discrete case r is integer, but in this paper $\frac{1}{2} \leq r \leq k - \frac{1}{2}$ can also be fractional. We will also sometimes refer to $r' = \frac{1}{2} - \theta$. Note that r' can be negative.

In the statistical setting, we refer to some fixed and unknown distribution D over instance-label pairs (X, Y), where we assume access to a sample (training set) drawn i.i.d. from D, and the objective is to minimize the expected loss:

$$\ell_\theta(w, D) = \mathbb{E}_{X,Y\sim D}[\ell_\theta(X, Y, w)]. \tag{3}$$

When the distribution D is clear from context, we simply write $\ell_\theta(w)$, and we might also omit the subscript θ. For fixed D and θ we let $w^* \in \operatorname{argmin}_{w\in\mathcal{H}_{k,\theta}} \mathbb{E}[\ell(X, Y, w)]$. This is the true minimizer of the loss on the distribution.

For a set of predictors (hypothesis class) H, we denote $\ell^*_\theta(H, D) := \min_{w\in H} \ell_\theta(w, D)$. For a sample $S \in ([0, 1]^d \times \{\pm 1\})^*$, we use the notation

$$\hat{\mathbb{E}}_S[f(X, Y)] = \frac{1}{|S|}\sum_{i=1}^{|S|} f(x_i, y_i) \tag{4}$$

and again sometimes drop the subscript S when it is clear from context. For a fixed sample S, and fixed θ and D, the empirical loss of a predictor w on the sample is denoted $\hat{\ell}(w) = \hat{\mathbb{E}}_S[\ell_\theta(X, Y, w)]$.

2.1 Rademacher Complexity

The empirical Rademacher complexity of the Winnow loss for a class $W \subseteq \mathbb{R}^d$ with respect to a sample $S = ((x_1, y_1), \ldots, (x_m, y_m)) \in ([0, 1]^d \times \{\pm 1\})^m$ is

$$\mathcal{R}(W, S) := \frac{2}{m}\mathbb{E}\left[\sup_{w\in W}\left|\sum_{i=1}^{m} \epsilon_i \ell(x_i, y_i, w)\right|\right] \tag{5}$$

where the expectation is over the *Rademacher random variables* $\epsilon_1, \ldots, \epsilon_m$. These are defined as independent random variables drawn uniformly from $\{\pm 1\}$. The average Rademacher complexity of the Winnow loss for a class $W \subseteq \mathbb{R}^d$ with respect to a distribution D over $[0, 1]^d \times \{\pm 1\}$ is denoted by

$$\mathcal{R}_m(W, D) := \mathbb{E}_{S\sim D^m}[\mathcal{R}(W, S)]. \tag{6}$$

We also define the average Rademacher complexity of W with respect to the *linear loss* by

$$\mathcal{R}_m^L(W, D) := \frac{2}{m}\mathbb{E}\left[\sup_{w\in W}\left|\sum_{i=1}^{m}\epsilon_i Y_i\langle w, X_i\rangle\right|\right] \tag{7}$$

where the expectation is over $\epsilon_1, \ldots, \epsilon_m$ as above and $((X_1, Y_1), \ldots, (X_m, Y_m)) \sim D^m$.

2.2 Probability Tools

We use the following variation on Bernstein's inequality.

Proposition 1 *Let $B > 0$. For a random variable $X \in [0, B]$, $\delta \in (0, 1)$ and n an integer, with probability at least $1 - \delta$ over n i.i.d. draws of X,*

$$\left|\hat{\mathbb{E}}[X] - \mathbb{E}[X]\right| \leq 2B\sqrt{\frac{\ln(1/\delta)}{n} \cdot \max\left\{\frac{\mathbb{E}[X]}{B}, \frac{\ln(1/\delta)}{n}\right\}}.$$

Proof By Bernstein's inequality (Bernstein, 1946), if $Z_1, \ldots, Z_n$ are i.i.d. draws from a random variable $Z \in [-1, 1]$ such that $\mathbb{E}[Z] = 0$, and $\mathrm{Var}[Z^2] = \sigma^2$, then

$$\mathbb{P}[\hat{\mathbb{E}}[Z] \geq \epsilon] \leq \exp\left(-\frac{n\epsilon^2}{2(\sigma^2 + \epsilon/3)}\right). \tag{8}$$

Fix $\delta \in (0, 1)$ and an integer n. If $\ln(1/\delta)/n \leq \sigma^2$ then let $\epsilon = 2\sqrt{\frac{\ln(1/\delta)}{n} \cdot \sigma^2} \leq 2\sigma^2$. In this case

$$\frac{n\epsilon^2}{2\sigma^2 + 2\epsilon/3} \geq \frac{n\epsilon^2}{10\sigma^2/3} \geq \ln(1/\delta).$$

If $\ln(1/\delta)/n > \sigma^2$ then let $\epsilon = 2\ln(1/\delta)/n$. Then $\sigma^2 \leq \ln(1/\delta)/n = \epsilon/2$. In this case

$$\frac{n\epsilon^2}{2\sigma^2 + 2\epsilon/3} \geq \frac{n\epsilon^2}{5\epsilon/3} \geq n\epsilon/4 = \ln(1/\delta).$$

In both cases, the RHS of Eq. (8) is at most δ. Therefore, with probability at least $1 - \delta$,

$$\hat{\mathbb{E}}[Z] \leq 2\sqrt{\frac{\ln(1/\delta)}{n}\max\left\{\sigma^2, \frac{\ln(1/\delta)}{n}\right\}}.$$

where the last inequality follows from the range of Z. Now, for a random variable X with range in $[0, B]$, let $Z = (X - \mathbb{E}[X])/B$. We have $\sigma^2 = \mathrm{Var}[Z] = \mathrm{Var}[X]/B^2 \leq \mathbb{E}[X^2/B^2] \leq \mathbb{E}[X/B]$, where the last inequality follows from the range of X. Therefore

$$\frac{\hat{\mathbb{E}}[X] - \mathbb{E}[X]}{B} \leq 2\sqrt{\frac{\ln(1/\delta)}{n}\max\left\{\frac{\mathbb{E}[X]}{B}, \frac{\ln(1/\delta)}{n}\right\}}.$$

The same bound on $\mathbb{E}[X] - \hat{\mathbb{E}}[X]$ can be derived similarly by considering $Z = (\mathbb{E}[X] - X)/B$. ■

We further use the following fact, which bounds the ratio between the empirical fraction of positive or negative labels and their true probabilities. We will apply this fact to make sure that enough negative and positive labels can be found in a random sample.

Proposition 2 *Let B be a binomial random variable, $B \sim Binomial(m,p)$. If $p \geq 8\ln(1/\delta)/m$ then with probability of at least $1-\delta$, $B \geq mp/2$.*

Proof This follows from a multiplicative Chernoff bound (Angluin and Valiant, 1979). ■

3. Online Algorithm

Consider the following algorithm:

Unnormalized Exponentiated Gradient (unnormalized-EG)

parameters: $\eta, \lambda > 0$
input: $z_1, \ldots, z_T \in \mathbb{R}^d$
initialize: $w_1 = (\lambda, \ldots, \lambda) \in \mathbb{R}^d$
update rule: $\forall i, w_{t+1}[i] = w_t[i] e^{-\eta z_t[i]}$

The following theorem provides a regret bound with local-norms for the unnormalized EG algorithm (for a proof, see Theorem 2.23 of Shalev-Shwartz, 2012).

Theorem 3 *Assume that the unnormalized EG algorithm is run on a sequence of vectors such that for all t, i we have $\eta z_t[i] \geq -1$. Then, for all $u \in \mathbb{R}^d_+$,*

$$\sum_{t=1}^{T} \langle w_t - u, z_t \rangle \leq \frac{d\lambda + \sum_{i=1}^{d} u[i] \ln(u[i]/(e\,\lambda))}{\eta} + \eta \sum_{t=1}^{T} \sum_{i=1}^{d} w_t[i] z_t[i]^2 \ .$$

Now, let us apply it to a case in which we have a sequence of convex functions $f_1, \ldots, f_T$, and z_t is the sub-gradient of f_t at w_t. Additionally, set $\lambda = k/d$ and consider u s.t. $\|u\|_1 \leq k$. We obtain the following.

Theorem 4 *Assume that the unnormalized EG algorithm is run with $\lambda = k/d$. Assume that for all t, we have $z_t \in \partial f_t(w_t)$, for some convex function f_t. Further assume that for all t, i we have $\eta z_t[i] \geq -1$, and that for some positive constants α, β, it holds that $\eta = \sqrt{k \ln(d)/(\beta T)}$, $T \geq 4\alpha^2 k \ln(d)/\beta$, and*

$$\sum_{i=1}^{d} w_t[i] z_t[i]^2 \leq \alpha f_t(w_t) + \beta \ . \tag{9}$$

Then, for all $u \in \mathbb{R}^d_+$, with $\|u\|_1 \leq k$ we have

$$\sum_{t=1}^{T} f_t(w_t) \leq \sum_{t=1}^{T} f_t(u) + \sqrt{\frac{4\alpha^2 k \ln(d)}{\beta T}} \cdot \sum_{t=1}^{T} f_t(u) + \sqrt{4\beta k \ln(d) T} + 4\alpha k \ln(d).$$

Proof Using the convexity of f_t and the assumption that $z_t \in \partial f_t(w_t)$ we have that

$$\sum_{t=1}^{T} (f_t(w_t) - f_t(u)) \leq \sum_{t=1}^{T} \langle w_t - u, z_t \rangle \ .$$

Combining with Theorem 3 we obtain

$$\sum_{t=1}^{T}(f_t(w_t) - f_t(u)) \leq \frac{d\lambda + \sum_{i=1}^{d} u[i]\ln(u[i]/(e\,\lambda))}{\eta} + \eta \sum_{t=1}^{T}\sum_{i=1}^{d} w_t[i] z_t[i]^2 \ .$$

Using the assumption in Eq. (9), the definition of $\lambda = k/d$, and the assumptions on u, we obtain

$$\sum_{t=1}^{T}(f_t(w_t) - f_t(u)) \leq \frac{k\ln(d)}{\eta} + \eta\beta T + \eta\alpha \sum_{t=1}^{T} f_t(w_t) \ .$$

Rearranging the above we conclude that

$$\sum_{t=1}^{T} f_t(w_t) \leq \frac{1}{1-\alpha\eta}\left(\sum_{t=1}^{T} f_t(u) + \frac{k\ln(d)}{\eta} + \eta\beta T\right).$$

Now, since $1/(1-x) \leq 1+2x$ for $x \in [0, 1/2]$ and $\alpha\eta \leq \frac{1}{2}$, we conclude, by substituting for the definition of η, that

$$\sum_{t=1}^{T} f_t(w_t) \leq \sum_{t=1}^{T} f_t(u) + 2\sqrt{k\ln(d)\beta T} + 2\alpha\sqrt{\frac{k\ln(d)}{\beta T}} \cdot \sum_{t=1}^{T} f_t(u) + 4\alpha k \ln(d).$$

■

We can now derive the desired regret bound for our algorithm. We also provide a bound for the statistical setting, using online-to-batch conversion.

Corollary 5 *Let $\ell \equiv \ell_\theta$ for some $\theta \in [0,k]$. Fix any sequence $(x_1,y_1),(x_2,y_2),\ldots,(x_T,y_T) \in [0,1]^d \times \{\pm 1\}$ and assume $T \geq 4k\ln(d)/r$. Suppose the unnormalized EG algorithm listed in Section 3 is run using $\eta := \sqrt{\frac{k\ln(d)}{rT}}$, $\lambda := k/d$, and any $z_t \in \partial_w \ell(x_t, y_t, w_t)$ for all t. Define $L_{\mathrm{UEG}} := \sum_{t=1}^{T} \ell(x_t, y_t, w_t)$, let $L(u) := \sum_{t=1}^{T} \ell(x_t, y_t, u)$, and let $u^* \in \operatorname{argmin} L(u)$. Then the following regret bound holds.*

$$L_{\mathrm{UEG}} - L(u^*) \leq \sqrt{16rk\ln(d)T} + 4k\ln(d). \tag{10}$$

Moreover, for $m \geq 1$, assume that a random sample $S = ((x_1,y_1),(x_2,y_2),\ldots,(x_m,y_m))$ is drawn i.i.d. from an unknown distribution D over $[0,1]^d \times \{\pm 1\}$. Then there exists an online-to-batch conversion of the UEG algorithm that takes S as input and outputs $\bar{w}$, such that

$$\mathbb{E}[\ell(\bar{w}, D)] \leq \ell(w^*, D) + \sqrt{\frac{16rk\ln(d)}{m}} + \frac{4k\ln(d)}{m}, \tag{11}$$

where the expectation is over the random draw of S.

Proof Every sub-gradient $z_t \in \partial_w \ell(x_t, y_t, w_t)$ is of the form $z_t = a_t x_t$ for some $a_t \in \{-1, 0, +1\}$. Since $0 \leq x_t[i] \leq 1$ and $w_t[i] \geq 0$ for all i, it follows that $\sum_{i=1}^{d} w_t[i] z_t[i]^2 = |a_t| \sum_{i=1}^{d} w[i] x_t[i]^2 \leq |a_t| \langle w_t, x_t \rangle$. Now consider three disjoint cases.

- Case 1: $\langle w_t, x_t \rangle \leq r$. Then $\sum_{i=1}^{d} w_t[i] z_t[i]^2 \leq \langle w_t, x_t \rangle \leq r$.

- Case 2: $\langle w_t, x_t\rangle > r$ and $y_t = 1$. Then $a_t = 0$ and $\sum_{i=1}^d w_t[i]z_t[i]^2 = 0$.
- Case 3: $\langle w_t, x_t\rangle > r$ and $y_t = -1$. Then $\sum_{i=1}^d w_t[i]z_t[i]^2 \le \langle w_t, x_t\rangle \le [r' + \langle w_t, x_t\rangle]_+ - r' \le [r' + \langle w_t, x_t\rangle]_+ + r$.

In all three cases, the final upper bound on $\sum_{i=1}^d w_t[i]z_t[i]^2$ is at most $\ell(x_t, y_t, w_t) + r$. Therefore, Eq. (9) from Theorem 4 is satisfied with $f_t(w) := \ell(x_t, y_t, w)$, $\alpha := 1$, and $\beta := r$. From Theorem 4 with this choice of f_t and the given settings of η, λ, and z_t, we get that for any u such that $\|u\|_1 \le k$,

$$L_{\mathrm{UEG}} \le L(u) + L(u)\sqrt{\frac{4k\ln(d)}{rT}} + \sqrt{4rk\ln(d)T} + 4k\ln(d). \tag{12}$$

Observing that $L(u^*) \le L(\mathbf{0}) \le rT$, we conclude the regret bound in Eq. (10).

For the statistical setting, a simple approach for online-to-batch conversion is to run the UEG algorithm as detailed in Corollary 5, with $T = m$, and to return the average predictor $\bar{w} = \frac{1}{m}\sum_{i\in[m]} w_i$. By standard analysis (e.g., Shalev-Shwartz, 2012, Theorem 5.1), $\mathbb{E}[\ell_\theta(\bar{w}, D)] \le \frac{1}{m}\mathbb{E}[L_{UEG}]$, where the expectation is over the random draw of S. Setting $u = w_*$, Eq. (12) gives

$$\mathbb{E}[\ell_\theta(\bar{w}, D)] \le \mathbb{E}\left[\hat{\ell}(w^*) + \sqrt{\hat{\ell}(w^*)^2 \cdot \frac{4k\ln(d)}{rm}} + \sqrt{\frac{4rk\ln(d)}{m}} + \frac{4k\ln(d)}{m}\right].$$

Since $\mathbb{E}[\hat{\ell}(w^*)] = \ell(w^*)$ and $\ell(w^*) \le r$, Eq. (11) follows. ■

In the online setting a simple version of the canonical mirror descent algorithm thus achieves the postulated regret bound of $O(\sqrt{rk\log(d)T}) \equiv O(\sqrt{\theta k\log(d)T})$. For the statistical setting, an online-to-batch conversion provides the desired rate of $O(rk\log(d)/\epsilon^2) \equiv O(\theta k\log(d)/\epsilon^2)$. Is this online-to-batch approach necessary, or is a similar rate for the statistical setting achievable also using standard ERM? Moreover, this online-to-batch approach leads to an improper algorithm, that is, the output w might not be in $\mathcal{H}_{k,\theta}$, since it might not satisfy the norm bound. In the next section we show that standard, proper, ERM, leads to the same learning rate.

4. ERM Upper Bound

We now proceed to analyze the performance of empirical risk minimization in the statistical batch setting. As above, assume a random sample $S = ((x_1, y_1), \ldots, (x_m, y_m))$ of pairs drawn i.i.d. according to a distribution D over $[0,1]^d \times \{\pm 1\}$. An empirical risk minimizer on the sample is denoted $\hat{w} \in \mathrm{argmin}_{w\in\mathcal{H}_{k,\theta}} \frac{1}{m}\sum_{i\in[m]} \ell(x_i, y_i, w)$. We wish to show an upper bound on $\ell(\hat{w}) - \ell(w^*)$. We will prove the following theorem:

Theorem 6 *For $k \ge r \ge 0$, and $m \ge k$, with probability $1-\delta$ over the random draw of S,*

$$\ell(\hat{w}) \le \ell(w^*) + \sqrt{\frac{O(rk(\ln(d)\ln^3(3m) + \ln(1/\delta)))}{m}} + \frac{O(r\log(1/\delta))}{m}. \tag{13}$$

The proof strategy is based on considering the loss on negative examples and the loss on positive examples separately. Denote

$$\ell_-(w, D) = \mathbb{E}_{(X,Y)\sim D}[\ell(X, Y, w) \mid Y = -1], \text{ and}$$
$$\ell_+(w, D) = \mathbb{E}_{(X,Y)\sim D}[\ell(X, Y, w) \mid Y = +1].$$

For a given sample, denote $\hat{\ell}_-(w) = \hat{\mathbb{E}}[\ell(X, Y, w) \mid Y = -1]$ and similarly for $\hat{\ell}_+(w)$. Denote $p_+ = \mathbb{E}_{(X,Y)\sim D}[Y = +1]$ and $\hat{p}_+ = \hat{\mathbb{E}}[Y = +1]$, and similarly for p_- and $\hat{p}_-$.

As Theorem 21 in Section 5 below shows, the rate of uniform convergence of $\hat{\ell}_-(w)$ to $\ell_-(w)$ for all $w \in \mathcal{H}_{k,\theta}$ is $\tilde{\Omega}(\sqrt{k^2/m})$, which is slower than the desired $\tilde{O}(\sqrt{\theta k/m})$. Therefore, uniform convergence analysis for $\mathcal{H}_{k,\theta}$ cannot provide a tight result. Instead, we define a subset $U_b \subseteq \mathcal{H}_{k,\theta}$, such that with probability at least $1 - \delta$, the empirical risk minimizer of a random sample is in U_b. We show that a uniform convergence rate of $\tilde{O}(\sqrt{\theta k/m})$ does in fact hold for all $w \in U_b$. The analysis of uniform convergence of the negative loss is carried out in Section 4.1.

For positive labels, uniform convergence rates over $\mathcal{H}_{k,\theta}$ in fact suffice to provide the desired guarantee. This analysis is provided in Section 4.2. The analysis uses the results in Section 3 for the online algorithm to construct a small cover of the relevant function class. This then bounds the Rademacher complexity of the class and leads to a uniform convergence guarantee. In Section 4.3, the two convergence results are combined, while taking into account the mixture of positive and negative labels in D.

4.1 Convergence on Negative Labels

We now commence the analysis for negative labels. Denote by D_- the distribution of $(X, Y) \sim D$ conditioned on $Y = -1$, so that $\mathbb{P}_{(X,Y)\sim D_-}[Y = -1] = 1$, and $\mathbb{P}_{(X,Y)\sim D_-}[X = x] = \mathbb{P}_{(X,Y)\sim D}[X = x \mid Y = -1]$. For $b \geq 0$ define

$$U_b(D) = \{w \in \mathbb{R}^d_+ \mid \|w\|_1 \leq k, \mathbb{E}_D[\langle w, X\rangle \mid Y = -1] \leq b\}.$$

Note that $U_b(D) \subseteq \mathcal{H}_{k,\theta}$.

We now bound the rate of convergence of $\hat{\ell}_-$ to ℓ_- for all $w \in U_b(D)$. We will then show that b can be set so that with high probability $\hat{w} \in U_b(D)$. Our technique is related to local Rademacher analysis (Bartlett et al., 2005), in that the latter also proposes to bound the Rademacher complexity of subsets of a function class, and uses these bounds to provide tighter convergence rates. Our analysis is better tailored to the Winnow loss, by taking into account the different effects of the negative and positive labels.

The convergence rate for $U_b(D)$ is bounded by first bounding $\mathcal{R}^L_m(U_b(D), D_-)$, the Rademacher complexity of the linear loss for the distribution over the examples with negative labels, and then concluding a similar bound on $\mathcal{R}_m(U_b(D), D)$. We start with a more general bound on $\mathcal{R}^L_m$.

Lemma 7 *For a fixed distribution over D over $[0, 1]^d \times \{\pm 1\}$, let $\alpha_j = \mathbb{E}_{(X,Y)\sim D}[X[j]]$, and let $\mu \in \mathbb{R}^d_+$. Define $U^\mu = \{w \in \mathbb{R}^d_+ \mid \langle w, \mu\rangle \leq 1\}$. Then if $dm \geq 3$,*

$$\mathcal{R}^L_m(U^\mu, D) \leq \max_{j:\alpha_j>0} \frac{1}{\mu_j} \sqrt{\frac{32 \ln(d)}{m} \cdot \max\left\{\alpha_j, \frac{\ln(dm)}{m}\right\}}$$

Proof Assume w.l.o.g that $\alpha_j > 0$ for all j (if this is not the case, dimensions with $\alpha_j = 0$ can be removed because this implies that $X[j] = 0$ with probability 1).

$$\begin{aligned}\frac{m}{2} R_m^L(U^\mu, S) &= \mathbb{E}_\sigma \left[\sup_{w:\langle w,\mu\rangle \le 1} \sum_{i=1}^m \sigma_i \langle w, x_i\rangle \right] \\ &= \mathbb{E}_\sigma \left[\sup_{w:\langle w,\mu\rangle \le 1} \langle w, \sum_{i=1}^m \sigma_i x_i\rangle \right] \\ &= \mathbb{E}_\sigma \left[\max_{j\in[d]} \sum_{i=1}^m \sigma_i \frac{x_i[j]}{\mu[j]} \right] .\end{aligned}$$

Therefore, using Massart's lemma (Massart, 2000, Lemma 5.2) and denoting $\hat{\alpha}_j = \frac{1}{m}\sum_{i\in[m]}^m x_i[j]$, we have:

$$\begin{aligned}R_m^L(U^\mu, S) &\le \frac{\sqrt{8\ln(d)}}{m} \cdot \max_j \frac{\sqrt{\sum_i x_i[j]^2}}{\mu[j]} \\ &\le \frac{\sqrt{8\ln(d)}}{m} \cdot \max_j \frac{\sqrt{\sum_i x_i[j]}}{\mu[j]} \\ &= \sqrt{\frac{8\ln(d)}{m}} \cdot \max_j \frac{\sqrt{\hat{\alpha}_j}}{\mu[j]} \\ &= \sqrt{\frac{8\ln(d)}{m} \cdot \max_j \frac{\hat{\alpha}_j}{\mu[j]^2}} .\end{aligned}$$

Taking expectation over S and using Jensen's inequality we obtain

$$R_m^L(U^\mu, D) = \mathbb{E}_S[R_m^L(U^\mu, S)] \le \sqrt{\frac{8\ln(d)}{m} \cdot \mathbb{E}_S[\max_j \frac{\hat{\alpha}_j}{\mu[j]^2}]}$$

By Bernstein's inequality (Proposition 1), with probability $1-\delta$ over the choice of $\{x_i\}$, for all $j \in [d]$

$$\hat{\alpha}_j \le \alpha_j + 2\sqrt{\frac{\ln(d/\delta)}{m} \cdot \max\left\{\alpha_j, \frac{\ln(d/\delta)}{m}\right\}}.$$

And, in any case, $\hat{\alpha}_j \le 1$. Therefore,

$$\mathbb{E}_S\left[\max_j \frac{\hat{\alpha}_j}{\mu[j]^2}\right] \le \max_j \frac{1}{\mu[j]^2}\left(\delta + \alpha_j + 2\sqrt{\frac{\ln(d/\delta)}{m} \cdot \max\left\{\alpha_j, \frac{\ln(d/\delta)}{m}\right\}}\right)$$

Choose $\delta = 1/m$ and let j be a maximizer of the above. Consider two cases. If $\alpha_j < \ln(dm)/m$ then

$$\mathbb{E}_S\left[\max_j \frac{\hat{\alpha}_j}{\mu[j]^2}\right] \le \max_j \frac{1}{\mu[j]^2} \cdot \frac{4\ln(dm)}{m}.$$

Otherwise,

$$\mathbb{E}_S\left[\max_j \frac{\hat{\alpha}_j}{\mu[j]^2}\right] \le \max_j \frac{1}{\mu[j]^2}(\delta + 3\alpha_j) \le \max_j \frac{4\alpha_j}{\mu[j]^2}.$$

All in all, we have shown

$$R_m^L(U^\mu, D) \leq \max_j \frac{1}{\mu[j]} \sqrt{\frac{32\ln(d)}{m} \cdot \max\left\{\alpha_j, \tfrac{\ln(dm)}{m}\right\}}.$$

■

The lemma above can now be used to bound the Rademacher complexity of the linear loss for D_-.

Lemma 8 *For any distribution D over $(X,Y) \in [0,1]^d \times \{\pm 1\}$, if $dm \geq 3$,*

$$\mathcal{R}_m^L(U_b(D), D_-) \leq \sqrt{\frac{128k\ln(d)}{m} \max\left\{b, \frac{k\ln(dm)}{m}\right\}}.$$

Proof Let $\alpha_j = \mathbb{E}_{(X,Y)\sim D_-}[X[j]]$. Let $J = \{j \in [d] \mid \alpha_j \geq \frac{b}{k}\}$, and $\bar{J} = \{j \in [d] \mid \alpha_j < \frac{b}{k}\}$. For a vector $v \in \mathbb{R}^d$ and a set $I \subseteq [d]$, denote by $v[I]$ the vector which is obtained from v by setting the coordinates not in I to zero. Let $((X_1,Y_1),\ldots,(X_m,Y_m)) \sim D_-^m$. By the definition of $\mathcal{R}_m^L$, with Rademacher random variables $\epsilon_1,\ldots,\epsilon_m$ (see Eq. 7), we have

$$\begin{aligned}
&\mathcal{R}_m^L(U_b(D), D_-) \\
&= \frac{2}{m}\mathbb{E}\left[\sup_{w\in U_b(D)}\left|\sum_{i=1}^m \epsilon_i Y_i \langle w, X_i\rangle\right|\right] \\
&= \frac{2}{m}\mathbb{E}\left[\sup_{w\in U_b(D)}\left|\sum_{i=1}^m \epsilon_i Y_i \langle w[J], X_i[J]\rangle + \sum_{i=1}^m \epsilon_i Y_i \langle w[\bar{J}], X_i[\bar{J}]\rangle\right|\right] \\
&\leq \frac{2}{m}\mathbb{E}\left[\sup_{w\in U_b(D)}\left|\sum_{i=1}^m \epsilon_i Y_i \langle w[J], X_i[J]\rangle\right|\right] + \frac{2}{m}\mathbb{E}\left[\sup_{w\in U_b(D)}\left|\sum_{i=1}^m \epsilon_i Y_i \langle w[\bar{J}], X_i[\bar{J}]\rangle\right|\right] \\
&= \mathcal{R}_m^L(U_b(D), D_1) + \mathcal{R}_m^L(U_b(D), D_2), \qquad (14)
\end{aligned}$$

where D_1 is the distribution of $(X[J], Y)$, where $(X,Y) \sim D_-$, and D_2 is the distribution of $(X[\bar{J}], Y)$. We now bound the two Rademacher complexities of the right-hand side using Lemma 7.

To bound $\mathcal{R}_m^L(U_b(D), D_1)$, define U^μ as in Lemma 7 for $\mu \in \mathbb{R}_+^d$, and define $\mu_1 \in \mathbb{R}_+^d$ by $\mu_1[j] = \alpha_j/b$. It is easy to see that $U_b(D) \subseteq U^{\mu_1}$. Therefore $\mathcal{R}_m^L(U_b(D), D_1) \leq \mathcal{R}_m^L(U^{\mu_1}, D_1)$. By Lemma 7 and the definition of μ_1

$$\begin{aligned}
\mathcal{R}_m^L(U^{\mu_1}) &\leq \max_{j\in J} \frac{1}{\mu_1[j]} \sqrt{\frac{32\ln(d)}{m} \max\left\{\alpha_j, \frac{\ln(dm)}{m}\right\}} \\
&= \max_{j\in J} \frac{b}{\alpha_j} \sqrt{\frac{32\ln(d)}{m} \max\left\{\alpha_j, \frac{\ln(dm)}{m}\right\}} \\
&= \max_{j\in J} \sqrt{\frac{b}{\alpha_j}\frac{32\ln(d)}{m} \max\left\{b, \frac{b}{\alpha_j}\frac{\ln(dm)}{m}\right\}}.
\end{aligned}$$

By the definition of J, for all $j \in J$ we have $\frac{b}{\alpha_j} \leq k$. It follows that

$$\mathcal{R}_m^L(U^{\mu_1}, D_1) \leq \sqrt{\frac{32k\ln(d)}{m} \max\left\{b, \frac{k\ln(dm)}{m}\right\}}. \tag{15}$$

To bound $\mathcal{R}_m^L(U_b(D), D_2)$, define $\mu_2 \in \mathbb{R}_+^d$ by $\mu_2[j] = \frac{1}{k}$. Note that $U^{\mu_2} = \mathcal{H}_{k,\theta}$ and $U_b(D) \subseteq \mathcal{H}_{k,\theta}$, hence $\mathcal{R}_m^L(U_b(D), D_2) \leq \mathcal{R}_m^L(U^{\mu_2}, D_2)$. By Lemma 7 and the definition of μ_2

$$\begin{aligned}\mathcal{R}_m^L(U^{\mu_2}, D_2) &\leq \max_{j\in\bar{J}} \frac{1}{\mu_2[j]} \sqrt{\frac{32\ln(d)}{m} \max\left\{\alpha_j, \frac{\ln(dm)}{m}\right\}} \\ &= \max_{j\in\bar{J}} \sqrt{\frac{32k\ln(d)}{m} \max\left\{k\alpha_j, \frac{k\ln(dm)}{m}\right\}}.\end{aligned}$$

By the definition of $\bar{J}$, for all $j \in J$ we have $k\alpha_j \leq b$. Therefore

$$\mathcal{R}_m^L(U^{\mu_2}, D_2) \leq \sqrt{\frac{32k\ln(d)}{m} \max\left\{b, \frac{k\ln(dm)}{m}\right\}}. \tag{16}$$

Combining Eq. (14), Eq. (15) and Eq. (16) we get the statement of the theorem. ■

Finally, the bound on $\mathcal{R}_m^L(U_b(D), D)$ is used in the following theorem to obtain a uniform convergence result of the negative loss for predictors in $U_b(D)$.

Theorem 9 *Let $b \geq 0$. There exists a universal constant C such that for any distribution D over $[0,1]^d \times \{\pm 1\}$, with probability $1-\delta$ over samples of size m, for any $w \in U_b(D)$,*

$$\ell_-(w) \leq \hat{\ell}_-(w) + C\left(\sqrt{\frac{kb\ln(d/\delta) + |r'|}{m\hat{p}_-}} + \frac{k\ln(dm\hat{p}_-/\delta)}{m\hat{p}_-}\right). \tag{17}$$

Proof Define $\phi : \mathbb{R} \to \mathbb{R}$ by $\phi(z) = [r' - z]_+$. Since $\mathbb{P}_{(X,Y)\sim D}[Y = -1] = 1$, the Winnow loss on pairs (X, Y) drawn from D is exactly $\phi(Y\langle w, X\rangle)$. Note that ϕ is an application of a 1-Lipschitz function to a translation of the linear loss. Thus, by the properties of the Rademacher complexity (Bartlett and Mendelson, 2002) and by Lemma 8 we have, for $dm \geq 3$,

$$\begin{aligned}\mathcal{R}_m(U_b(D), D_-) &\leq \mathcal{R}_m^L(U_b(D), D_-) \\ &\leq \sqrt{\frac{128k\ln(d)}{m} \max\left\{b, \frac{k\ln(dm)}{m}\right\}}.\end{aligned} \tag{18}$$

Assume that $r' \leq 0$. By Talagrand's inequality (see, e.g., Boucheron et al., 2005, Theorem 5.4), with probability $1-\delta$ over samples of size m drawn from D_-, for all $w \in U_b(D)$

$$\ell(w) \leq \hat{\ell}(w) + 2\mathcal{R}_m(U_b(D), D_-) + \sqrt{\frac{2\sup_{w\in U_b(D)} \mathrm{Var}_{D_-}[\ell(X,Y,w)]\ln(1/\delta)}{m}} + \frac{4k\ln(1/\delta)}{3m}. \tag{19}$$

To bound $\mathrm{Var}_{D_-}[\ell(X,Y,w)]$, note that $\ell(X,Y,w) \in [0,k]$. In addition, $\mathbb{P}_{D_-}[Y=-1]=1$, thus with probability 1, $\ell(X,Y,w) = [r' + \langle w, X\rangle]_+ \leq \langle w, x\rangle$, where the last inequality follows from the assumption $r' \leq 0$. Therefore, for any $w \in U_b(D)$

$$\mathrm{Var}_{D_-}[\ell(X,Y,w)] \leq \mathbb{E}[\ell^2(X,Y,w)] \leq \mathbb{E}_{D_-}[k\ell(X,Y,w)] \leq k \cdot \mathbb{E}_{D_-}[\langle w, X\rangle] \leq kb. \tag{20}$$

Combining Eq. (18), Eq. (19) and Eq. (20) we conclude that there exists a universal constant C such that for any $w \in U_b(D)$, if a sample of size m is drawn i.i.d. from D_-, then

$$\ell(w) \leq \hat{\ell}(w) + C\left(\sqrt{\frac{kb\ln(d/\delta)}{m}} + \frac{k\ln(dm/\delta)}{m}\right).$$

If $r' > 0$, $\hat{\ell}_-(w) - \ell_-(w)$ is identical to the case $r' = 0$, thus the same result holds.

To get Eq. (17), consider a sample of size m drawn from D instead of D_-. In this case, $\ell(w, D_-) = \ell_-(w, D)$, $\hat{\ell}(w, D_-) = \hat{\ell}_-(w, D)$, and the effective sample size for D_- is $m\hat{p}_-$. ■

We now show that with an appropriate setting of b, $\hat{w} \in U_b(D)$ with high probability over the draw of a sample from D. First, the following lemma provides a sample-dependent guarantee for $\hat{w}$.

Lemma 10 *Let $\hat{w}$ and $\hat{p}_-$ be defined as above and let $\hat{E} := \hat{E}_S$ for the fixed sample S defined above. Then*

$$\hat{\mathbb{E}}[\langle \hat{w}, X\rangle \mid Y=-1] \leq \frac{r}{\hat{p}_-}.$$

Proof Let $m_+ = |\{i \mid y_i = +1\}|$, and $m_- = |\{i \mid y_i = -1\}|$. By the definition of the hinge function and the fact that $\langle x_i, \hat{w}\rangle \geq 0$ for all i we have that

$$\begin{aligned} m_- r' + \sum_{y_i=-1} \langle x_i, \hat{w}\rangle &\leq \sum_{y_i=-1} (r' + \langle x_i, \hat{w}\rangle) \\ &\leq \sum_{y_i=+1} [r - \langle x_i, \hat{w}\rangle]_+ + \sum_{y_i=-1} [r' + \langle x_i, \hat{w}\rangle]_+ \\ &= \sum_{i\in[m]} \ell(x_i, y_i, \hat{w}). \end{aligned}$$

By the optimality of $\hat{w}$, $\sum_{i\in[m]} \ell(x_i, y_i, \hat{w}) \leq \sum_{i\in[m]} \ell(x_i, y_i, \mathbf{0}) = m_+ r + m_- [r']_+$. Therefore

$$\sum_{y_i=-1} \langle x_i, \hat{w}\rangle \leq m_+ r + m_-([r']_+ - r') = m_+ r + m_-[-r']_+ \leq (m_+ + m_-)r = mr,$$

where we have used the definitions of r' and r to conclude that $[-r']_+ \leq r$. Dividing both sides by m_- we conclude our proof. ■

The following lemma allows converting the sample-dependent restriction on $\hat{w}$ given in Lemma 10 to one that holds with high probability over samples.

Lemma 11 *For any distribution over $[0,1]^d$, with probability $1-\delta$ over samples of size n, for any $w \in \mathcal{H}_{k,\theta}$*

$$\mathbb{E}[\langle w, X\rangle] \leq 2\hat{\mathbb{E}}[\langle w, X\rangle] + \frac{16k\ln(\frac{d}{\delta})}{n}.$$

Proof For every $j \in [d]$, denote $\alpha_j = \mathbb{E}[X[j]]$. Denote $\hat{\alpha}_j = \hat{\mathbb{E}}[X[j]]$. By Bernstein's inequality (Proposition 1), with probability $1-\delta$,

$$\alpha_j \leq \hat{\alpha}_j + 2\sqrt{\frac{\ln(1/\delta)}{n} \cdot \max\left\{\alpha_j, \frac{\ln(1/\delta)}{n}\right\}} \leq \hat{\alpha}_j + \max\left\{\frac{\alpha_j}{2}, \frac{8\ln(1/\delta)}{n}\right\},$$

where the last inequality can be verified by considering the cases $\alpha_j \leq \frac{16\ln(1/\delta)}{n}$ and $\alpha_j \geq \frac{16\ln(1/\delta)}{n}$. Applying the union bound over $j \in [d]$ we obtain that with probability of $1-\delta$ over samples of size n, for any $w \in \mathcal{H}_{k,\theta}$

$$\begin{aligned}\mathbb{E}[\langle w, X\rangle] = \langle w, \alpha\rangle &\leq \sum_{j\in[d]} w_j \left(\hat{\alpha}_j + \frac{\alpha_j}{2} + \frac{8\ln(d/\delta)}{n}\right) \\ &\leq \hat{\mathbb{E}}[\langle w, X\rangle] + \frac{1}{2}\mathbb{E}[\langle w, X\rangle] + \frac{8\ln(d/\delta)}{n} \cdot k.\end{aligned}$$

Thus $\mathbb{E}[\langle w, X\rangle] \leq 2\hat{\mathbb{E}}\langle w, X\rangle + \frac{16k\ln(d/\delta)}{n}$. ■

Combining the two lemmas above, we conclude that with high probability, $\hat{w} \in U_b$ for an appropriate setting of b.

Lemma 12 *If $p_- \geq \frac{8\ln(1/\delta)}{m}$, then with probability $1-\delta$ over samples of size m, $\hat{w} \in U_b(D)$, where*

$$b = \frac{4r}{p_-} + \frac{32k\ln(2d/\delta)}{mp_-}. \tag{21}$$

Proof Apply Lemma 11 to D_-. With probability of $1-\delta$ over samples of size n drawn from D_-,

$$\mathbb{E}_{D_-}[\langle w, X\rangle] \leq 2\hat{\mathbb{E}}_{D_-}[\langle w, X\rangle] + \frac{16k\ln(d/\delta)}{n}.$$

Now, consider a sample of size m drawn according to D. Then $\mathbb{E}_{D_-}[\cdot] = \mathbb{E}_D[\cdot \mid Y=-1]$, and $n = m\hat{p}_-$. Therefore, with probability $1-2\delta$,

$$\begin{aligned}\mathbb{E}[\langle w, X\rangle \mid Y=-1] &\leq 2\hat{\mathbb{E}}[\langle w, X\rangle \mid Y=-1] + \frac{16k\ln(d/\delta)}{m\hat{p}_-} \\ &\leq \frac{2r}{\hat{p}_-} + \frac{16k\ln(d/\delta)}{m\hat{p}_-} \\ &\leq \frac{4r}{p_-} + \frac{32k\ln(d/\delta)}{mp_-},\end{aligned} \tag{22}$$

where the second inequality follows from Lemma 10, and the last inequality follows from the assumption on p_- and Proposition 2. ■

This lemma shows that to bound the sample complexity of an ERM algorithm for the Winnow loss, it suffices to bound the convergence rates of the empirical loss for $w \in U_b(D)$, with b defined as in Eq. (21). Thus, we will be able to use Theorem 9 to bound the convergence of the loss on negative examples.

4.2 Convergence on Positive Labels

For positive labels, we show a uniform convergence result that holds for the entire class $\mathcal{H}_{k,\theta}$. The idea of the proof technique below is as follows. First, following a technique in the spirit of the one given by Zhang (2002), we show that the regret bound for the online learning algorithm presented in Section 3 can be used to construct a small cover of the set of loss functions parameterized by $\mathcal{H}_{k,\theta}$. Second, we convert the bound on the size of the cover to a bound on the Rademacher complexity, thus showing a uniform convergence result. This argument is a refinement of Dudley's entropy bound (Dudley, 1967), which is stated in explicit terms by Srebro et al. (2010, Lemma A.3).

We first observe that by Theorem 4, if the conditions of the theorem hold and there is u such that $f_t(u) = 0$ for all t, then

$$\frac{1}{T}\sum_{t=1}^{T} f_t(w_t) \leq 4\sqrt{\frac{\beta k \ln(d)}{T}}. \tag{23}$$

Let $k \geq r \geq 0$ be two real numbers and let $W \subseteq \mathbb{R}^d_+$. Let ϕ_w denote the function defined by $\phi_w(x, y) = \ell(x, y, w)$, and consider the class of functions $\Phi_W = \{\phi_w \mid w \in W\}$. Given $S = ((x_1, y_1), \ldots, (x_m, y_m))$, where $x_i \in [0, 1]^d$ and $y_i \in \{\pm 1\}$, we say that (Φ_W, S) is (∞, ϵ)-properly-covered by a set $V \subseteq \Phi_W$ if for any $f \in \Phi_W$ there is a $g \in V$ such that

$$\|(f(x_1, y_1), \ldots, f(x_m, y_m)) - (g(x_1, y_1), \ldots, g(x_m, y_m))\|_\infty \leq \epsilon.$$

We denote by $\mathbb{N}_\infty(W, S, \epsilon)$ the minimum value of an integer N such that exists a $V \subseteq \Phi_W$ of size N that (∞, ϵ)-properly-covers (Φ_W, S).

The following lemma bounds the covering number for F_W, for sets S with all-positive labels y_i.

Lemma 13 *Let $S = ((x_1, 1), \ldots, (x_m, 1))$, where $x_i \in [0, 1]^d$. Then,*

$$\ln \mathbb{N}_\infty(\mathcal{H}_{k,\theta}, S, \epsilon) \leq 16 \cdot rk \ln(d) \ln(3m)/\epsilon^2.$$

Proof We use a technique in the spirit of the one given by Zhang (2002). Fix some u, with $u \geq 0$ and $\|u\|_1 \leq k$. For each i let

$$g_i^u(w) = \begin{cases} |\langle w, x_i\rangle - \langle u, x_i\rangle| & \text{if } \langle u, x_i\rangle \leq r \\ [r - \langle w, x_i\rangle]_+ & \text{o.w.} \end{cases}$$

and define the function

$$G_u(w) = \max_i g_i^u(w) \ .$$

It is easy to verify that for any w,

$$\|(\phi_w(x_1, 1), \ldots, \phi_w(x_m, 1)) - (\phi_u(x_1, 1), \ldots, \phi_u(x_m, 1))\|_\infty \leq G_u(w).$$

Now, clearly, $G_u(u) = 0$. In addition, for any $w \geq 0$, a sub-gradient of G_u at w is obtained by choosing i that maximizes $g_i^u(w)$ and then taking a sub-gradient of g_i^u, which is of the form $z = \alpha x_i$ where $\alpha \in \{-1, 0, 1\}$. If $\alpha \in \{-1, 1\}$, it is easy to verify that

$$\sum_j w[j] z[j]^2 \leq \langle w, x_i\rangle \leq g_i^u(w) + r = G_u(w) + r \ .$$

If $\alpha = 0$ then clearly $\sum_j w[j]z[j]^2 \leq G_u(w) + r$ as well.

We can now use Eq. (23) by setting $f_t = G_u$ for all t, setting $\alpha = 1$ and $\beta = r$ in Eq. (9), and noting that since $x_i \in [0,1]^d$, we have $z_t \in [-1,1]^d$ for all t. If $\eta \leq 1$ we have $\eta z_t[i] \geq -1$ for all t, i as needed. Since $\eta = \sqrt{\frac{k\ln(d)}{rT}}$, this holds for all $T \geq k\ln(d)/r$.

We conclude that if we run the unnormalized EG algorithm with $T \geq k\ln(d)/r$ and η and λ as required, we get

$$\sum_{t=1}^{T} G_u(w_t) \leq 4\sqrt{rk\ln(d)T}.$$

Dividing by T and using Jensen's inequality we conclude

$$G_u\left(\tfrac{1}{T}\sum_t w_t\right) \leq 4\sqrt{\frac{rk\ln(d)}{T}}.$$

Denote $w_u = \frac{1}{T}\sum_t w_t$. Setting $\epsilon = 4\sqrt{\frac{rk\ln(d)}{T}}$, it follows that the following set is a (∞,ϵ)-proper-cover for $(F_{\mathcal{H}_{k,\theta}}, S)$:

$$V = \{w_u \mid u \in \mathcal{H}_{k,\theta}\}.$$

Now, we only have left to bound the size of V. Consider again the unnormalized EG algorithm. Since $z_t = \alpha x_i$ for some $\alpha \in \{-1,0,+1\}$ and $i \in \{1,\ldots,m\}$, at each round of the algorithm there are only two choices to be made: the value of i and the value of α. Therefore, the number of different vectors produced by running unnormalized EG for T iterations on G_u for different values of u is at most $(3m)^T$. Thus $|V| \leq (3m)^T$. By our definition of ϵ,

$$\ln|V| \leq T\ln(3m) \leq 16rk\ln(d)\ln(3m)/\epsilon^2.$$

This concludes our proof. ■

Using this result we can bound from above the covering number defined using the Euclidean norm: We say that (Φ_W, S) is $(2,\epsilon)$-properly-covered by a set $V \subseteq \Phi_W$ if for any $f \in \Phi_W$ there is a $g \in V$ such that

$$\frac{1}{\sqrt{m}}\|(f(x_1,y_1),\ldots,f(x_m,y_m)) - (g(x_1,y_1),\ldots,g(x_m,y_m))\|_2 \leq \epsilon.$$

We denote by $\mathbb{N}_2(W,S,\epsilon)$ the minimum value of an integer N such that exists a $V \subseteq \Phi_W$ of size N that $(2,\epsilon)$-properly-covers (Φ_W, S). It is easy to see that for any two vectors $u, v \in \mathbb{R}^m$, $\frac{1}{\sqrt{m}}\|u-v\|_2 \leq \|u-v\|_\infty$. It follows that for any W and S, we have $\mathbb{N}_2(W,S,\epsilon) \leq \mathbb{N}_\infty(W,S,\epsilon)$.

The $\mathbb{N}_2$ covering number can be used to bound the Rademacher complexity of (Φ_W, S) using a refinement of Dudley's entropy bound (Dudley, 1967), which is stated explicitly by Srebro et al. (2010, Lemma A.3). The lemma states that for any $\epsilon \geq 0$,

$$\mathcal{R}(W,S) \leq 4\epsilon + \frac{10}{\sqrt{m}}\int_\epsilon^B \sqrt{\ln \mathbb{N}_2(W,S,\gamma)}\,d\gamma,$$

where B is an upper bound on the possible values of $f \in \Phi_W$ on members of S. For S with all-positive labels we clearly have $B \leq r$.

Combining this with Lemma 13, we get

$$\mathcal{R}(\mathcal{H}_{k,\theta}, S) \leq C \cdot \left(\epsilon + \frac{1}{\sqrt{m}} \int_{\epsilon}^{r} \sqrt{rk \ln(d) \ln(3m)}/\gamma \, d\gamma \right) = C \cdot \left(\epsilon + \sqrt{\frac{rk \ln(d) \ln(3m)}{m}} \ln(r/\epsilon) \right).$$

Setting $\epsilon = rk/m$ we get

$$\mathcal{R}(\mathcal{H}_{k,\theta}, S) \leq C \cdot \sqrt{\frac{rk \ln(d) \ln^3(3m)}{m}}.$$

Thus, for any distribution D over $[0,1]^d \times \{\pm 1\}$ that draws only positive labels, we have

$$\mathcal{R}_m(\mathcal{H}_{k,\theta}, D) \leq C \left(\sqrt{\frac{rk \ln(d) \ln^3(3m)}{m}} \right).$$

By Rademacher sample complexity bounds (Bartlett and Mendelson, 2002), and since ℓ for positive labels is bounded by r, we can immediately conclude the following:

Theorem 14 *Let $k \geq r \geq 0$. For any distribution D over $[0,1]^d \times \{\pm 1\}$ that draws only positive labels, with probability $1 - \delta$ over samples of size m, for any $w \in \mathcal{H}_{k,\theta}$,*

$$\begin{aligned} \ell_+(w) &\leq \hat{\ell}_+(w) + C \cdot \left(\sqrt{\frac{rk \ln(d) \ln^3(3m)}{m}} + \sqrt{\frac{r^2 \ln(1/\delta)}{m}} \right) \\ &\leq \hat{\ell}_+(w) + C \cdot \left(\sqrt{\frac{rk(\ln(d) \ln^3(3m) + \ln(1/\delta))}{m}} \right). \end{aligned}$$

4.3 Combining Negative and Positive Losses

We have shown separate convergence rate results for the loss on positive labels and for the loss on negative labels. We now combine these results to achieve a convergence rate upper bound for the full Winnow loss. To do this, the convergence results given above must be adapted to take into account the fraction of positive and negative labels in the true distribution as well as in the sample. The following theorems accomplish this for the negative and the positive cases. First, a bound is provided for the positive part of the loss.

Theorem 15 *There exists a universal constant C such that for any distribution D over $[0,1]^d \times \{\pm 1\}$, with probability $1 - \delta$ over samples of size m*

$$p_+ \ell_+(\hat{w}) \leq \hat{p}_+ \hat{\ell}_+(\hat{w}) + C \cdot \sqrt{\frac{rk(\ln(kd) \ln^3(m) + \ln(3/\delta))}{m}}.$$

Proof First, if $p_+ \leq \frac{8\ln(1/\delta)}{m}$ then the theorem trivially holds. Therefore we assume that $p_+ \geq \frac{8\ln(1/\delta)}{m}$. We have

$$p_+\ell_+(\hat{w}) = \hat{p}_+\hat{\ell}_+(\hat{w}) + (p_+ - \hat{p}_+)\hat{\ell}_+(\hat{w}) + p_+(\ell_+(\hat{w}) - \hat{\ell}_+(\hat{w})). \tag{24}$$

To prove the theorem, we will bound the two rightmost terms. First, to bound $(p_+ - \hat{p}_+)\hat{\ell}_+(\hat{w})$, note that by definition of the loss function for positive labels we have that $\hat{\ell}_+(\hat{w}) \in [0, r]$. Therefore, Bernstein's inequality (Proposition 1) implies that with probability $1-\delta/3$

$$(p_+ - \hat{p}_+)\hat{\ell}_+(\hat{w}) \leq 2r\sqrt{\frac{\ln(3/\delta)}{m}\max\left\{p_+, \frac{\ln(3/\delta)}{m}\right\}} \leq \sqrt{\frac{4r\ln(3/\delta)}{m}}. \tag{25}$$

Second, to bound $p_+(\ell_+(\hat{w}) - \hat{\ell}_+(\hat{w}))$, we apply Theorem 14 to the conditional distribution induced by D on X given $Y = 1$, to get that with probability $1 - \delta/3$

$$p_+(\ell_+(\hat{w}) - \hat{\ell}_+(\hat{w})) \leq p_+ \cdot C \cdot \sqrt{\frac{rk(\ln(d)\ln^3(3m) + \ln(3/\delta))}{m\hat{p}_+}}.$$

Using our assumption on p_+ we obtain from Proposition 2 that with probability $1 - \delta/3$, $p_+/\hat{p}_+ \leq 2$. Therefore, $p_+/\sqrt{\hat{p}_+} \leq \sqrt{2p_+} \leq \sqrt{2}$. Thus, with probability $1 - 2\delta/3$,

$$p_+(\ell_+(\hat{w}) - \hat{\ell}_+(\hat{w})) \leq C \cdot \sqrt{\frac{rk(\ln(d)\ln^3(3m) + \ln(3/\delta))}{m}}. \tag{26}$$

Combining Eq. (24), Eq. (25) and Eq. (26) and applying the union bound, we get the theorem. ■

Second, a bound is provided for the negative part of the loss.

Theorem 16 *There exists a universal constant C such that for any distribution D over $[0,1]^d \times \{\pm 1\}$, with probability $1 - \delta$ over samples of size m*

$$p_-\ell_-(\hat{w}) \leq \hat{p}_-\hat{\ell}_-(\hat{w}) + C\left(\sqrt{\frac{rk\ln(d/\delta)}{m}} + \frac{k\ln(dm/\delta)}{m}\right). \tag{27}$$

Proof First, if $p_- \leq \frac{8\ln(1/\delta)}{m}$ then the theorem trivially holds (since $\ell_-(\hat{w}) \in [0, r+k]$). Therefore we assume that $p_- \geq \frac{8\ln(1/\delta)}{m}$. Thus, by Proposition 2, $\hat{p}_- \geq p_-/2$. We have

$$p_-\ell_-(\hat{w}) = \hat{p}_-\hat{\ell}_-(\hat{w}) + (p_- - \hat{p}_-)\hat{\ell}_-(\hat{w}) + p_-(\ell_-(\hat{w}) - \hat{\ell}_-(\hat{w})). \tag{28}$$

To prove the theorem, we will bound the two rightmost terms. First, to bound $(p_- - \hat{p}_-)\hat{\ell}_-(\hat{w})$, note that by Bernstein's inequality (Proposition 1) and our assumption on p_-, with probability $1 - \delta$

$$p_- - \hat{p}_- \leq 2\sqrt{\frac{\ln(1/\delta)}{m}\max\left\{p_-, \frac{\ln(1/\delta)}{m}\right\}} = 2\sqrt{\frac{p_-\ln(1/\delta)}{m}}.$$

By Lemma 10 and Proposition 2, $\hat{\ell}_-(\hat{w}) \leq \frac{2r}{\hat{p}_-} \leq \frac{4r}{p_-}$. In addition, by definition $\hat{\ell}_-(\hat{w}) \leq r + k \leq 2k$. Therefore

$$(p_- - \hat{p}_-)\hat{\ell}_-(\hat{w}) \leq 4 \min\left\{\frac{2r}{p_-}, k\right\} \sqrt{\frac{p_- \ln(1/\delta)}{m}}. \tag{29}$$

Now, if $k > 2r/p_-$, then the right-hand of the above becomes

$$8\frac{r}{p_-}\sqrt{\frac{p_- \ln(1/\delta)}{m}} = 8\sqrt{\frac{(r/p_-) \cdot r \ln(1/\delta)}{m}} \leq 8\sqrt{\frac{k \cdot r \ln(1/\delta)}{m}} .$$

Otherwise, $k \leq 2r/p_-$ and the right-hand of Eq. (29) becomes

$$4k\sqrt{\frac{p_- \ln(1/\delta)}{m}} \leq 4k\sqrt{\frac{(2r/k) \ln(1/\delta)}{m}} \leq 8\sqrt{\frac{k \cdot r \ln(1/\delta)}{m}} .$$

All in all, we have shown that

$$(p_- - \hat{p}_-)\hat{\ell}_-(\hat{w}) \leq 8\sqrt{\frac{rk \ln(1/\delta)}{m}}. \tag{30}$$

Second, to bound $p_-(\ell_-(\hat{w}) - \hat{\ell}_-(\hat{w}))$, recall that by Lemma 12, we have $\hat{w} \in U_b(D)$, where

$$b = \frac{4r}{p_-} + \frac{32k \ln(d/\delta)}{mp_-} \leq \frac{C}{p_-}\left(2r + \frac{k \ln(d/\delta)}{m}\right).$$

Thus, by Theorem 9, with probability $1 - \delta$

$$\ell_-(w) \leq \hat{\ell}_-(w) + C\left(\sqrt{\frac{kb \ln(d/\delta)}{m\hat{p}_-}} + \frac{k \ln(dm/\delta)}{m\hat{p}_-}\right).$$

Since $\hat{p}_- \geq p_-/2$,

$$\ell_-(w) \leq \hat{\ell}_-(w) + C\left(\sqrt{\frac{kb \ln(d/\delta)}{mp_-}} + \frac{k \ln(dm/\delta)}{mp_-}\right).$$

for some other constant C. Therefore, substituting b for its upper bound we get

$$p_-(\ell_-(w) - \hat{\ell}_-(w)) \leq C\left(\sqrt{\frac{kr \ln(d/\delta)}{m}} + \frac{k \ln(dm/\delta)}{m}\right). \tag{31}$$

Combining Eq. (28), Eq. (30) and Eq. (31) we get the statement of the theorem. ■

Finally, we prove our main result for the sample complexity of ERM algorithms for Winnow.

Proof (Proof of Theorem 6) From Theorem 15 and Theorem 16 we conclude that with probability $1-\delta$,

$$\begin{aligned}\ell(\hat{w}) &= p_-\ell_-(\hat{w}) + p_+\ell_+(\hat{w}) \\ &\leq \hat{p}_-\hat{\ell}_-(\hat{w}) + \hat{p}_+\hat{\ell}_+(\hat{w}) + \sqrt{\frac{O(rk(\ln(d)\ln^3(3m)+\ln(1/\delta)))}{m}}. \end{aligned} \tag{32}$$

Now,

$$\hat{p}_-\hat{\ell}_-(\hat{w}) + \hat{p}_+\hat{\ell}_+(\hat{w}) = \hat{\ell}(\hat{w}) \leq \hat{\ell}(w^*). \tag{33}$$

We have $\mathbb{E}[\ell(X,Y,w^*)] = \ell(w^*) \leq \ell(\mathbf{0}) \leq r$. By Bernstein's inequality (Proposition 1), with probability $1-\delta$

$$\begin{aligned}\hat{\ell}(w^*) = \hat{\mathbb{E}}[\ell(X,Y,w^*)] &\leq \mathbb{E}[\ell(X,Y,w^*)] + 2r\sqrt{\frac{\ln(1/\delta)}{m}\max\left\{\frac{\mathbb{E}[\ell(X,Y,w^*)]}{r}, \frac{\ln(1/\delta)}{m}\right\}} \\ &\leq \ell(w^*) + 2\sqrt{\frac{r^2\ln(1/\delta)}{m}} + 2\frac{r\ln(1/\delta)}{m}.\end{aligned}$$

Combining this with Eq. (33), we get that with probability $1-\delta$

$$\hat{p}_-\hat{\ell}_-(\hat{w}) + \hat{p}_+\hat{\ell}_+(\hat{w}) \leq \ell(w^*) + 2\sqrt{\frac{r^2\ln(1/\delta)}{m}} + 2\frac{r\ln(1/\delta)}{m}.$$

In light of Eq. (32), we conclude Eq. (13) ■

Theorem 6 shows that using empirical risk minimization, the loss of the obtained predictor converges to the loss of the optimal predictor at a rate of the order

$$\tilde{O}\left(\sqrt{\frac{rk\log(d)}{m}}\right) \equiv \tilde{O}\left(\sqrt{\frac{\theta k\log(d)}{m}}\right).$$

Up to logarithmic factors, this is the best possible rate for learning in the generalized Winnow setting. This is shown in the next section, in Theorem 17. We also show, in Theorem 21, that this rate cannot be obtain via standard uniform convergence analysis.

5. Lower Bounds

In this section we provide lower bounds for the learning rate and for the uniform convergence rate of the Winnow loss ℓ_θ.

5.1 Learning Rate Lower Bound

Fix a threshold θ. The best Winnow loss for a distribution D over $[0,1]^d \times \{\pm 1\}$ using a hyperplane from a set $W \subseteq \mathbb{R}^d_+$ is denoted by $\ell^*_\theta(W) = \min_{w\in W}\ell_\theta(w)$. The following result shows that even if the data domain is restricted to the discrete domain $\{0,1\}^d$, the number of samples required for learning with the Winnow loss grows at least linearly in θk. This resolves an open question posed by Littlestone (1988).

Theorem 17 *Let $k \geq 1$ and let $\theta \in [1, k/2]$. The sample complexity of learning $\mathcal{H}_{k,\theta}$ with respect to the loss ℓ_θ is $\Omega(\theta k/\epsilon^2)$. That is, for all $\epsilon \in (0, 1/2)$ if the training set size is $m = o(\theta k/\epsilon^2)$, then for any learning algorithm, there exists a distribution such that the classifier, $h : \{0,1\}^d \to \mathbb{R}_+$, that the algorithm outputs upon receiving m i.i.d. examples satisfies $\ell_\theta(h) - \ell^*_\theta(\mathcal{H}_{k,\theta}) > \epsilon$ with a probability of at least $1/4$.*

The construction which shows the lower bound proceeds in several stages: First, we prove that there exists a set of size k^2 in $\{\pm 1\}^{k^2}$ which is shattered on the linear loss with respect to predictors with a norm bounded by k. Then, apply a transformation on this construction to show a set in $\{0,1\}^{2k^2+1}$ which is shattered on the linear loss with a threshold of $k/2$. In the next step, we adapt the construction to hold for any value of the threshold. Finally, we use the resulting construction to prove Theorem 17.

The construction uses the notion of a *Hadamard matrix*. A Hadamard matrix of order n is an $n \times n$ matrix H_n with entries in $\{\pm 1\}$ such that $H_n H_n^T = n I_n$. In other words, all rows in the matrix are orthogonal to each other. Hadamard matrices exist at least for each n which is a power of 2 (Sylvester, 1867). The first lemma constructs a shattered set for the linear loss on $\{\pm 1\}^{k^2}$.

Lemma 18 *Assume k is a power of 2, and let $d = k^2$. Let $x_1, \ldots, x_d \subseteq \{\pm 1\}^d$ be the rows of the Hadamard matrix of order d. For every $y \in \{\pm 1\}^d$, there exists a $w \in W' = \{w \in [-1,1]^d \mid \|w\| \leq k\}$ such that for all $i \in [d]$, $y[i]\langle w, x_i\rangle = 1$.*

Proof By the definition of a Hadamard matrix, for all $i \neq j$, $\langle x_i, x_j\rangle = 0$. Given $y \in \{\pm 1\}^d$, set $w = \frac{1}{d}\sum_{j\in[d]} y_j x_j$. Then for each i,

$$y_i\langle w, x_i\rangle = y_i \frac{1}{d}\sum_{j\in[d]} y_j\langle x_i, x_j\rangle = \frac{1}{d} y_i^2 \langle x_i, x_i\rangle = \frac{1}{d}\|x_i\|_2^2 = 1.$$

It is left to show that $w \in W'$. First, for all $i \in [d]$, we have

$$|w[i]| = |\frac{1}{d}\sum_{j\in[d]} y_j x_j[i]| \leq \frac{1}{d}\sum_{j\in[d]} |x_j[i]| = 1,$$

which yields $w \in [-1,1]^d$. Second, using $\|w\|_1 \leq \sqrt{d}\|w\|_2$ and

$$\|w\|_2^2 = \langle w, w\rangle = \frac{1}{d^2}\sum_{i,j\in[d]} \langle y_i x_i, y_j x_j\rangle = \frac{1}{d^2}\sum_{i\in[d]} y_i^2\langle x_i, x_i\rangle = \frac{1}{d^2}\sum_{i\in[d]} d = 1,$$

we obtain that $\|w\|_1 \leq \sqrt{d} = k$. ■

The next lemma transforms the construction from Lemma 18 to a linear loss with a threshold of $k/2$.

Lemma 19 *Let k be a power of 2 and let $d = 2k^2 + 1$. There is a set $\{x_1, \ldots, x_{k^2}\} \subseteq \{0,1\}^d$ such that for every $y \in \{\pm 1\}^{k^2}$, there exists $w \in \mathcal{H}_{k,\theta}$ such that for all $i \in [k^2]$, $y[i](\langle w, x_i\rangle - k/2) = \frac{1}{2}$.*

Proof From Lemma 18 we have that there is a set $X = \{x_1, \ldots, x_{k^2}\} \subseteq \{\pm 1\}^{k^2}$ such that for each labeling $y \in \{\pm 1\}^{k^2}$, there exists a $w_y \in [-1, 1]^d$ with $\|w_y\|_1 \leq k$ such that for all $i \in [k^2]$, $y[i]\langle w_y, x_i\rangle = 1$. We now define a new set $\tilde{X} = \{\tilde{x}_1, \ldots, \tilde{x}_{k^2}\} \subseteq \{0,1\}^d$ based on X that satisfies the requirements of the lemma.

For each $i \in [k^2]$ let $\tilde{x}_i = [\frac{\vec{1}+x_i}{2}, \frac{\vec{1}-x_i}{2}, 1]$, where $[\cdot,\cdot,\cdot]$ denotes a concatenation of vectors and $\vec{1}$ is the all-ones vector. In words, each of the first k^2 coordinates in $\tilde{x}_i$ is 1 if the corresponding coordinate in x_i is 1, and zero otherwise. Each of the next k^2 coordinates in $\tilde{x}_i$ is 1 if the corresponding coordinate in x_i is -1, and zero otherwise. The last coordinate in $\tilde{x}_i$ is always 1.

Now, let $y \in \{\pm 1\}^{k^2}$ be a desired labeling. We defined $\tilde{w}_y$ based on w_y as follows: $\tilde{w}_y = [[w_y]_+, [-w_y]_+, \frac{k-\|w_y\|_1}{2}]$, where by $z = [v]_+$ we mean that $z[j] = \max\{v[j], 0\}$. In words, the first k^2 coordinates of $\tilde{w}_y$ are copies of the positive coordinates of w_y, with zero in the negative coordinates, and the next k^2 coordinates of $\tilde{w}_y$ are the absolute values of the negative coordinates of w_y, with zero in the positive coordinates. The last coordinate is a scaling term.

We now show that $\tilde{w}_y$ has the desired property on $\tilde{X}$. For each $i \in [k^2]$,

$$\begin{aligned}\langle \tilde{w}_y, \tilde{x}_i\rangle &= \left\langle \frac{\vec{1}+x_i}{2}, [w_y]_+ \right\rangle + \left\langle \frac{\vec{1}-x_i}{2}, [-w_y]_+ \right\rangle + \frac{k-|w_y|_1}{2} \\ &= \frac{|w_y|_1}{2} + \frac{\langle x_i, w_y\rangle}{2} + \frac{k-|w_y|_1}{2} = \frac{\langle x_i, w_y\rangle}{2} + \frac{k}{2} = \frac{y_i}{2} + \frac{k}{2}.\end{aligned}$$

It follows that $y_i(\langle \tilde{w}_y, \tilde{x}_i\rangle - k/2) = y_i^2/2 = 1/2$.

Now, clearly $\tilde{w}_y \in \mathbb{R}_+^d$. In addition,

$$\|\tilde{w}_y\|_1 = \|w_y\|_1 + \frac{k-\|w_y\|_1}{2} = \frac{\|w_y\|_1}{2} + \frac{k}{2} \leq k.$$

Hence $\tilde{w}_y \in \mathcal{H}_{k,\theta}$ as desired. ■

The last lemma adapts the previous construction to hold for any threshold.

Lemma 20 *Let z be a power of 2 and let k such that z divides k. Let $d = 2kz + k/z$. There is a set $\{x_1, \ldots, x_{zk}\} \subseteq \{0,1\}^d$ such that for every $y \in \{\pm 1\}^{zk}$, there exists a $w \in \mathcal{H}_{k,\theta}$ such that for all $i \in [zk]$, $y[i](\langle w, x_i\rangle - z/2) = \frac{1}{2}$.*

Proof By Lemma 19 there is a set $X = \{x_1, \ldots, x_{z^2}\} \subseteq \{0,1\}^{2z^2+1}$ such that for all $y \in \{\pm 1\}^{z^2}$, there exists a $w_y \in \mathbb{R}_+^{2z^2+1}$ such that $\|w_y\|_1 \leq z$ and for all $i \in [z^2]$, $y[i](\langle w_y, x_i\rangle - z/2) = \frac{1}{2}$.

We now construct a new set $\tilde{X} = \{\tilde{x}_1, \ldots, \tilde{x}_{zk}\} \subseteq \{0,1\}^{2kz+k/z}$ as follows: For $i \in [zk]$, let $n = \lfloor i/z^2 \rfloor$ and $m = i \mod z^2$, so that $i = nz^2 + m$.The vector $\tilde{x}_i$ is the concatenation of $\frac{kz}{z^2} = \frac{k}{z}$ vectors, each of which is of dimension $2z^2+1$, where all the vectors are the all-zeros vector, except the $(n+1)$'th vector which equals to x_{m+1}. That is:

$$\tilde{x}_i = [\overbrace{0}^{\in \mathbb{R}^{2z^2+1}}, \ldots, \overbrace{0}^{\in \mathbb{R}^{2z^2+1}}, \overbrace{x_{m+1}}^{\text{block } n+1}, \overbrace{0}^{\in \mathbb{R}^{2z^2+1}}, \ldots, \overbrace{0}^{\in \mathbb{R}^{2z^2+1}}] \in \mathbb{R}^{\frac{k}{z}(2z^2+1)}.$$

Given $\tilde{y} \in \{\pm 1\}^{kz}$, let us rewrite it as a concatenation of k/z vectors, each of which in $\{\pm 1\}^{z^2}$, namely,

$$\tilde{y} = [\overbrace{\tilde{y}(1)}^{\in\{\pm 1\}^{z^2}}, \ldots, \overbrace{\tilde{y}(k/z)}^{\in\{\pm 1\}^{z^2}}] \in \{\pm 1\}^{kz} .$$

Define $\tilde{w}_{\tilde{y}}$ as the concatenation of k/z vectors in $\{\pm 1\}^{z^2}$, using w_y defined above for each $y \in \{\pm 1\}^{z^2}$, as follows:

$$\tilde{w}_{\tilde{y}} = [\overbrace{w_{\tilde{y}(1)}}^{\in\mathbb{R}_+^{2z^2+1}}, \ldots, \overbrace{w_{\tilde{y}(k/z)}}^{\in\mathbb{R}_+^{2z^2+1}}] \in \mathbb{R}_+^{\frac{k}{z}(2z^2+1)} .$$

For each i such that $n = \lfloor i/z^2 \rfloor$ and $m = i \mod z^2$, we have

$$\langle \tilde{w}_{\tilde{y}}, \tilde{x}_i \rangle - z/2 = \langle w_{\tilde{y}(n+1)}, x_{m+1} \rangle - z/2 = \frac{1}{2}\tilde{y}(n+1)[m+1].$$

Now $\tilde{y}(n+1)[m+1] = \tilde{y}[i]$, thus we get $\tilde{y}[i](\langle \tilde{w}_{\tilde{y}}, \tilde{x}_i \rangle - z/2) = \frac{1}{2}$ as desired. Finally, we observe that $\|\tilde{w}_{\tilde{y}}\|_1 = \sum_{n\in[k/z]} \|w_{\tilde{y}(n)}\|_1 \leq k/z \cdot z = k$, hence $\tilde{w}_{\tilde{y}} \in \mathcal{H}_{k,\theta}$. ■

Finally, the construction above is used to prove the convergence rate lower bound.

Proof (Proof of Theorem 17) Let $k \geq 1$, $\theta \in [\frac{1}{2}, \frac{k}{2}]$. Define $z = 2\theta$. Let $n = \max\{n \mid 2^n \leq z\}$, and let $m = \max\{m \mid m2^n \leq k\}$. Define $\tilde{z} = 2^n$ and $\tilde{k} = m2^n$. We have that $\tilde{z}$ is a power of 2 and $\tilde{z}$ divides $\tilde{k}$. Let $\tilde{d} = 2\tilde{k}\tilde{z} + \tilde{k}/\tilde{z}$. By Lemma 20, there is a set $X = \{x_1, \ldots, x_{\tilde{z}\tilde{k}}\} \subseteq \{0,1\}^{\tilde{d}}$ such that for every $y \in \{\pm 1\}^{|X|}$, there exists a $w_y \in \mathcal{H}_{k,\theta}$ such that for all $i \in [\tilde{z}\tilde{k}]$, $y[i](\langle w_y, x_i \rangle - \tilde{z}/2) = \frac{1}{2}$.

Now, let $d = \tilde{d} + 1$, and define $\tilde{w}_y = [w_y, \frac{z-\tilde{z}}{2}]$ and $\tilde{x}_i = [x_i, 1]$. It follows that

$$\begin{aligned} y[i](\langle \tilde{w}_y, \tilde{x}_i \rangle - \theta) &= y[i](\langle \tilde{w}_y, \tilde{x}_i \rangle - z/2) \\ &= y[i](\langle w_y, x_i \rangle + z/2 - \tilde{z}/2 - z/2) \\ &= y[i](\langle w_y, x_i \rangle - \tilde{z}/2) = \frac{1}{2}. \end{aligned}$$

We conclude that for all $i \in [\tilde{z}\tilde{k}]$, $\ell_\theta(\tilde{x}_i, y[i], \tilde{w}_y) = 0$ and $\ell_\theta(\tilde{x}_i, 1 - y[i], \tilde{w}_y) = 1$. Moreover, $\text{sign}(\langle \tilde{w}_y, \tilde{x}_i \rangle - \theta) = y[i]$.

Now, for a given w define $h_w(x) = \text{sign}(\langle w, x_i \rangle - \theta)$, and consider the binary hypothesis class $H = \{h_w \mid w \in \mathcal{H}_{k,\theta}\}$ over the domain X. Our construction of $\tilde{w}_y$ shows that the set X is shattered by this hypothesis class, thus its VC dimension is at least $|X|$. By VC-dimension lower bounds (e.g., Anthony and Bartlett, 1999, Theorem 5.2), it follows that for any learning algorithm for H, if the training set size is $o(|X|/\epsilon^2)$, then there exists a distribution over X so that with probability greater than $1/64$, the output $\hat{h}$ of the algorithm satisfies

$$\mathbb{E}[\hat{h}(x) \neq y] > \min_{w\in\mathcal{H}_{k,\theta}} \mathbb{E}[h_w(x) \neq y] + \epsilon . \tag{34}$$

Next, we show that the existence of a learning algorithm for $\mathcal{H}_{k,\theta}$ with respect to ℓ_θ whose sample complexity is $o(|X|/\epsilon^2)$ would contradict the above statement. Indeed, let w^* be a minimizer of the right-hand side of Eq. (34), and let y^* be the vector of predictions

of w^* on X. As our construction of $\tilde{w}_{y^*}$ shows, we have $\ell_\theta(\tilde{w}_{y^*}) = \mathbb{E}[h_{w^*}(x) \neq y]$. Now, suppose that some algorithm learns $\hat{w} \in \mathcal{H}_{k,\theta}$ so that $\ell_\theta(\hat{w}) \leq \ell^*_\theta(\mathcal{H}_{k,\theta}) + \epsilon$. This implies that

$$\ell_\theta(\hat{w}) \leq \ell_\theta(\tilde{w}_{y^*}) + \epsilon = \mathbb{E}[h_{w^*}(x) \neq y] + \epsilon \ .$$

In addition, define a (probabilistic) classifier, $\hat{h}$, that outputs the label $+1$ with probability $p(\hat{w}, x)$ where $p(\hat{w}, x) = \min\{1, \max\{0, 1/2 + (\langle \hat{w}, x\rangle - \theta)\}\}$. Then, it is easy to verify that

$$\mathbb{P}[\hat{h}(x) \neq y] \leq \ell_\theta(x, y, \hat{w}) \ .$$

Therefore, $\mathbb{E}[\hat{h}(x) \neq y] \leq \ell_\theta(\hat{w})$, and we obtain that

$$\mathbb{E}[\hat{h}(x) \neq y] \leq \mathbb{E}[h_{w^*}(x) \neq y] + \epsilon \ ,$$

which leads to the desired contradiction. ■

We next show that the uniform convergence rate for our problem is in fact slower than the achievable learning rate.

5.2 Uniform Convergence Lower Bound

The next theorem shows that the rate of uniform convergence for our problem is asymptotically slower than the rate of convergence of the empirical loss minimizer given in Theorem 6, even if the drawn label in a random pair is negative with probability 1. This indicates that indeed, a more subtle argument than uniform convergence is needed to show that ERM learns at a rate of $\tilde{O}(\sqrt{\theta k/m})$, as done in Section 4.

Theorem 21 *Let $k \geq 1$, and assume $\theta \leq k/2$. There exists a distribution D over $\{0,1\}^{k^2+1} \times Y$ such that $\forall x \in \{0,1\}^d, \mathbb{P}[Y = -1 \mid X = x] = 1$, and $\ell^*(\mathcal{H}_{k,\theta}, D) = [r']_+$, and such that with probability at least $1/2$ over samples $S \sim D^m$,*

$$\exists w \in \mathcal{H}_{k,\theta}, \quad |\ell(w, S) - \ell(w, D)| \geq \Omega(\sqrt{k^2/m}). \tag{35}$$

This claim may seem similar to well-known uniform convergence lower bounds for classes with a bounded VC dimension (see, e.g., Anthony and Bartlett, 1999, Chapter 5). However, these standard results rely on constructions with non-realizable distributions, while Theorem 21 asserts the existence of a realizable distribution which exhibits this lower bound.

To prove this theorem we first show two useful lemmas. The first lemma shows that a lower bound on the uniform convergence of a function class can be derived from a lower bound on the Rademacher complexity of a related function class.

Lemma 22 *Let Z be a set, and consider a function class $F \subseteq [0,1]^Z$. Let D be a distribution over Z. Let $\bar{F} = \{(x_1, x_2) \to f(x_1) - f(x_2) \mid f \in F\}$. With probability at least $1 - \delta$ over samples $S \sim D^m$,*

$$\exists f \in F, \quad |\mathbb{E}_{X \sim S}[f(X)] - \mathbb{E}_{X \sim D}[f(X)]| \geq \frac{1}{4}\mathcal{R}_m(\bar{F}, D \times D) - \sqrt{\frac{\ln(1/\delta)}{8m}}. \tag{36}$$

Proof Denote $E[f,S] = \mathbb{E}_{X\sim S}[f(X)]$, and $E[f,D] = \mathbb{E}_{X\sim D}[f(X)]$. Consider two independent samples $S = (X_1,\ldots,X_m), S' = (X'_1,\ldots,X'_m) \sim D^m$. Let $\sigma = (\sigma_1,\ldots,\sigma_m)$ be Rademacher random variables, and let $S \sim (D\times D)^m$. We have

$$\begin{aligned}
2\cdot\mathbb{E}_S\left[\sup_{f\in F}|E[f,S]-E[f,D]|\right] &= \mathbb{E}_{S,S'}\left[\sup_{f\in F}|E[f,S]-E[f,D]| + \sup_{f\in F}|E[f,S']-E[f,D]|\right] \\
&\geq \mathbb{E}_{S,S'}\left[\sup_{f\in F}|E[f,S]-E[f,D]| + |E[f,S']-E[f,D]|\right] \\
&\geq \mathbb{E}_{S,S'}\left[\sup_{f\in F}|E[f,S]-E[f,S']|\right] \\
&= \frac{1}{m}\mathbb{E}_{S,S'}\left[\sup_{f\in F}\left|\sum_{i\in[m]} f(X_i)-f(X'_i)\right|\right] \\
&= \frac{1}{m}\mathbb{E}_{\sigma,\bar{S}}\left[\sup_{\bar{f}\in\bar{F}}\left|\sum_{i\in[m]}\sigma_i\bar{f}(X_i)\right|\right] = \mathcal{R}_m(\bar{F}, D\times D)/2.
\end{aligned}$$

We have left to show a lower bound with high probability. Define $g(S) = \sup_{f\in F}|E[f,S] - E[f,D]|$. Any change of one element in S can cause $g(S)$ to change by at most $1/m$, Therefore, by McDiarmid's inequality, $\mathbb{P}[g(S) \leq \mathbb{E}[g(S)] - t] \leq \exp(-2mt^2)$. Eq. (36) thus holds with probability $1-\delta$. ■

The next lemma provides a uniform convergence lower bound for a universal class of binary functions.

Lemma 23 *Let $H = \{0,1\}^{[n]}$ be the set of all binary functions on $[n]$. Let D be the uniform distribution over $[n]$. For any $n \geq 45$ and $m \geq 32n$, with probability of at least $\frac{1}{2}$ over i.i.d. samples of size m drawn from D,*

$$\exists h\in H, \quad |\mathbb{E}_{X\sim S}[h(X)] - \mathbb{E}_{X\sim D}[h(X)]| \geq \sqrt{\frac{n}{512m}}.$$

Proof Let $n \geq 45$ and $m \geq 32n$. By Lemma 22, it suffices to provide a lower bound for $\mathcal{R}_m(\bar{H}, D\times D)$. Fix a sample $S = ((x_1,x'_1),\ldots,(x_m,x'_m)) \sim (D\times D)^m$. We have

$$\frac{m}{2}\mathcal{R}(\bar{H},S) = \mathbb{E}_\sigma\left[\left|\sup_{h\in H}\sum_{i=1}^m \sigma_i(h(x_i)-h(x'_i))\right|\right],$$

where $\sigma = (\sigma_1, \ldots, \sigma_m)$ are Rademacher random variables. For a given $\sigma \in \{\pm 1\}^m$, define $h_\sigma \in H$ such that $h_\sigma(j) = \text{sign}(\sum_{i:x_i=j} \sigma_i - \sum_{i:x'_i=j} \sigma_i)$. Then

$$\begin{aligned}\frac{m}{2}\mathcal{R}(\bar{H}, S) &\geq \mathbb{E}_\sigma \left[\left| \sum_{i\in[m]} \sigma_i (h_\sigma(x_i) - h_\sigma(x'_i)) \right| \right] \\ &= \mathbb{E}_\sigma \left[\left| \sum_{j\in[n]} \left(\sum_{i:x_i=j} \sigma_i - \sum_{i:x'_i=j} \sigma_i \right) h_\sigma(j) \right| \right] \\ &= \sum_{j\in[n]} \mathbb{E}_\sigma \left[\left| \sum_{i:x_i=j} \sigma_i - \sum_{i:x'_i=j} \sigma_i \right| \right].\end{aligned}$$

Let $c_j(S)$ be the number of indices i such that exactly one of $x_i = j$ and $x'_i = j$ holds. Then $\mathbb{E}_\sigma[|\sum_{i:x_i=j} \sigma_i - \sum_{i:x'_i=j} \sigma_i|]$ is the expected distance of a random walk of length $c_j(S)$, which can be bounded from below by $\sqrt{c_j(S)/2}$ (Szarek, 1976). Therefore,

$$\mathcal{R}(\bar{H}, S) \geq \frac{\sqrt{2}}{m} \sum_{j\in[n]} \sqrt{c_j(S)}.$$

Taking expectation over samples, we get

$$\mathcal{R}(\bar{H}, D \times D) = \mathbb{E}_{S\sim(D\times D)^m}[\mathcal{R}(\bar{H}, S)] \geq \frac{\sqrt{2}}{m} \sum_{j\in[n]} \mathbb{E}_S \left[\sqrt{c_j(S)} \right]. \tag{37}$$

Our final step is to bound $\mathbb{E}_S \left[\sqrt{c_j(S)} \right]$. We have

$$\mathbb{E}_S[c_j(S)] = m \left(\frac{1}{n} - \frac{1}{n^2} \right) \geq \frac{m}{2n},$$

and

$$\text{Var}_S[c_j(S)] = m \left(\frac{1}{n} - \frac{1}{n^2} \right) \left(1 - \frac{1}{n} + \frac{1}{n^2} \right) \leq \frac{m}{n}.$$

Thus, by Chebyshev's inequality,

$$\mathbb{P} \left[c_j(S) \leq \frac{m}{2n} - t \right] \leq \frac{m}{nt^2}.$$

Therefore

$$\mathbb{E}_S \left[\sqrt{c_j(S)} \right] \geq \left(1 - \frac{m}{nt^2} \right) \sqrt{\frac{m}{2n} - t}.$$

Setting $t = \frac{m}{4n}$, and since $m/n \geq 32$, $\mathbb{E}_S \left[\sqrt{c_j(S)} \right] \geq \sqrt{\frac{m}{16n}}$. Plugging this into Eq. (37), we get that $\mathcal{R}(\bar{H}, D \times D) \geq \sqrt{\frac{n}{8m}}$. By Lemma 22, it follows that with probability at least $1 - \delta$ over samples,

$$\exists f \in F, \quad |\mathbb{E}_{X\sim S}[f(X)] - \mathbb{E}_{X\sim D}[f(X)]| \geq \sqrt{\frac{n}{128m}} - \sqrt{\frac{\ln(1/\delta)}{8m}}.$$

Fixing $\delta = 1/2$, we get that since $n \geq 64\ln(2)$, the RHS is at least $\sqrt{\frac{n}{512m}}$. ■

Using the two lemmas above, we are now ready to prove our uniform convergence lower bound. This is done by mapping a subset of $\mathcal{H}_{k,\theta}$ to a universal class of binary functions over $\Theta(k^2)$ elements from our domain. Note that for this lower bound it suffices to consider the more restricted domain of binary vectors.

Proof (Proof of Theorem 21) Let q be the largest power of 2 such that $q \leq k$. By Lemma 19, there exists a set of vectors $Z = \{z_1, \ldots, z_{q^2}\} \subseteq \{0,1\}^{q^2+1}$ such that for every $t \in \{\pm 1\}^{q^2}$ there exists a $w_t \in \mathcal{H}_{k,\theta}$ such that for all i, $t[i](\langle w, z_i\rangle - q/2) = \frac{1}{2}$. Denote $U = \{w_t \mid t \in \{\pm 1\}^{q^2}\}$. It suffices to prove a lower bound on the uniform convergence of U, since this implies the same lower bound for $\mathcal{H}_{k,\theta}$. Define the distribution D over $Z \times \{\pm 1\}$ such that for $(X, Y) \sim D$, X is drawn uniformly from $z_1, \ldots, z_{q^2}$ and $Y = -1$ with probability 1.

Consider the set of functions $H = \{0,1\}^Z$, and for $h \in H$ define $t_h \in \{\pm 1\}^{q^2}$ such that for all $i \in [q^2]$, $t_h[i] = 2h(z_i) - 1$. For any $i \in q^2$, we have

$$\ell(z_i, -1, w_{t_h}) = [r' + \langle w, z_i\rangle]_+ = [r' + (t[i]+k)/2]_+ = [r' + (k-1)/2 + h(i)]_+ = r' + (k-1)/2 + h(z_i).$$

The last equality follows since $r' \geq \frac{1-k}{2}$. It follows that for any $h \in H$ and any sample S drawn from D,

$$|\ell(w_{t_h}, S) - \ell(w_{t_h}, D)| = |\mathbb{E}_{X\sim S}[h(X)] - \mathbb{E}_{X\sim D}[h(X)]|.$$

By Lemma 23, with probability of at least $\frac{1}{2}$ over the sample $S \sim D^m$,

$$\exists h \in H, \quad |\mathbb{E}_{X\sim S}[h(X)] - \mathbb{E}_{X\sim D}[h(X)]| \geq \Omega(\sqrt{q^2/m}) = \Omega(\sqrt{k^2/m}).$$

Thus, with probability at least $1/2$,

$$\exists w \in \mathcal{H}_{k,\theta}, \quad |\ell(w_{t_h}, S) - \ell(w_{t_h}, D)| \geq \Omega(\sqrt{k^2/m}).$$

■

Acknowledgements

Tong Zhang is supported by the following grants: NSF IIS1407939, NSF IIS1250985, and NIH R01AI116744.

References

D. Angluin and L. G. Valiant. Fast probabilistic algorithms for Hamiltonian circuits and matchings. *Journal of Computer and System Sciences*, 18(2):155–193, April 1979.

M. Anthony and P. L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999.

P. Auer and M.K. Warmuth. Tracking the best disjunction. *Machine Learning*, 32(2): 127–150, 1998.

P. L. Bartlett and S. Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002.

P. L. Bartlett, O. Bousquet, and S. Mendelson. Local rademacher complexities. *Annals of Statistics*, 33(4):1497–1537, 2005.

S. Bernstein. *The Theory of Probabilities*. Gastehizdat Publishing House, Moscow, 1946.

S. Boucheron, O. Bousquet, and G. Lugosi. Theory of classification: a survey of recent advances. *ESAIM: Probability and Statistics*, 9:323–375, 2005.

R.M. Dudley. The sizes of compact subsets of hilbert space and continuity of gaussian processes. *Journal of Functional Analysis*, 1(3):290 – 330, 1967.

C. Gentile. The robustness of the p-norm algorithms. *Machine Learning*, 53:265–299, 2003.

S. M. Kakade, K. Sridharan, and A. Tewari. On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. In *Proceedings of NIPS*, 2009.

J. Kivinen and M. Warmuth. Additive versus exponentiated gradient updates for learning linear functions. Technical Report UCSC-CRL-94-16, University of California Santa Cruz, Computer Research Laboratory, 1994.

N. Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2:285–318, 1988.

N. Littlestone. *Mistake Bounds and Logarithmic Linear-Threshold Learning Algorithms*. PhD thesis, U. C. Santa Cruz, March 1989.

P. Massart. Some applications of concentration inequalities to statistics. In *Annales de la Faculté des Sciences de Toulouse*, volume 9:2, pages 245–303. Université Paul Sabatier, 2000.

R.E. Schapire, Y. Freund, P. Bartlett, and W.S. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. In *Machine Learning: Proceedings of the Fourteenth International Conference*, pages 322–330, 1997. To appear, *The Annals of Statistics*.

S. Shalev-Shwartz. *Online Learning: Theory, Algorithms, and Applications*. PhD thesis, The Hebrew University, 2007.

S. Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2012.

N. Srebro, K. Sridharan, and A. Tewari. Smoothness, low-noise and fast rates. *CoRR*, abs/1009.3896, 2010.

N. Srebro, K. Sridharan, and A. Tewari. On the universality of online mirror descent. *Advances in Neural Information Processing Systems (NIPS)*, 2011.

J.J. Sylvester. Thoughts on inverse orthogonal matrices, simultaneous signsuccessions, and tessellated pavements in two or more colours, with applications to newton's rule, ornamental tile-work, and the theory of numbers. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 34(232):461–475, 1867.

S.J. Szarek. On the best constants in the Khinchin inequality. *Studia Math*, 58(2), 1976.

T. Zhang. Covering number bounds of certain regularized linear function classes. *Journal of Machine Learning Research*, 2:527–550, 2002.